PHD

Condition monitoring of fluid power systems using artificial neural networks

Hsu, Cheng-Yu

Award date:
1995

Awarding institution:
University of Bath

Condition monitoring of fluid power systems using artificial neural networks

submitted by Cheng-Yu Hsu

for the degree of PhD

of the University of Bath

October, 1995

Summary

In this research, the feasibility and advantages of using artificial intelligence related techniques, such as artificial neural networks and fuzzy logic, for monitoring hydraulic power systems were investigated. Fundamental concepts related to artificial neural networks, fuzzy logic and condition monitoring were also reviewed.

A sequential hydraulic rig was built to test the artificial neural networks and a multi-layer back-propagation network was trained to model this system. The results showed that the network can accurately model the system. Later, a variety of faults were introduced into the system and the patterns of errors were used to train three different types of neural networks to encode these fault patterns. These neural networks were subsequently tested using untrained error patterns. A range of different hierarchical structures were proposed for the networks. All three types of neural networks were tested to be suitable for identifying faults. However, the adaptive resonance theory network was considered to be the most suitable network for fault identification. A multi-layer back-propagation network was also trained to model a real transmission rig. Once again, the network was found to predict system outputs with accuracy.

Fuzzy logic and reasoning methods for condition monitoring were also investigated. A simulation transmission system was simulated using a software package developed at the Fluid Power Centre at University of Bath, and was used for testing the technique derived from fuzzy logic. In this test, a multi-layer back-propagation network was trained to model the transmission system, and then prediction errors generated from this neural model were used for fault detection and identification. From the simulation results, it is considered that this method is one of the most promising techniques for condition monitoring.

Acknowledgements

There are many that have helped me to complete this work in many different ways. I would like to thank my parents, my wife, my sister and my brothers, for their encouragement and financial support. Also, I would like to acknowledge with appreciation the guidance by Dr. Tilley and Dr. Darling. Thanks are due to my government, the Republic of China in Taiwan, for funding this work. Most importantly, I would like to thank Dr. Y-H Young, Defence General Hospital in Taipei, Dr. Bennett, Medical Centre of Bath University, doctors and nurses of Royal United Hospital in Bath for saving my life and continuously monitoring my health condition during the past two years.

Contents

Chapter 1
Introduction

1.1 The need for condition monitoring systems

Condition monitoring is an action in which indicating or measuring instruments are applied for characterising the states of components in a system. Fault diagnosis is the process of identifying the root cause of a malfunction. Sometimes there is a little confusion with the terminology of condition monitoring and fault diagnosis. Theoretically, condition monitoring and fault diagnosis have different meanings and distinctive goals, the former for detecting faults and the latter for identifying faults. Nevertheless, in application and in literature the term condition monitoring is frequently used to cover the functions of condition monitoring and fault diagnosis. Although condition monitoring is not a new subject to either engineers or researchers, the importance of this field of engineering has become more demanding in recent years. Benefited by the massive applications of computers and related advanced technologies, numerous highly efficient, but very complex and expensive, machines are used in modern industries. It is predictable that the growth in the adoption of computerised equipment in the near future will be faster than ever. The introduction of high-technology equipment, present difficulties for technicians and engineers as it is difficult for them to monitor the operating conditions and identify malfunctions. The reason for this is due to the combination of lack of experience and the complexity of the machines. In addition, the capital and operating costs of modern systems is usually high and users can not afford the losses of frequently shutting them down for regular checks or allowing them to be idle during breakdowns. Furthermore, safety regulations are becoming tighter and any fault that could cause injury or even death must be predicted and prevented before any catastrophic failure occurs. Therefore, an advanced, automatic and reliable condition monitoring system which is capable of issuing warning signals and identifying a fault with accuracy and speed, is highly desirable.

Prevention is better than cure. With the technological explosion since 1970, the concept of predictive maintenance[1,2], which is based on condition monitoring, can be helpful in cutting unnecessary costs due to inefficient maintenance procedures. Rao[3] pointed out that a recent Department of Trade and Industry survey estimated potential benefits as much as £1.6 billion per year in manufacturing industries as a result of adopting predictive maintenance practices. In the same article, he also mentioned that implementing effective and efficient monitoring strategies could save 20% of the national fuel consumption bill per year in the UK. Also the industrial maintenance study suggested that a modest 5% increase in machine availability could secure a 30% profitability improvement for some companies[4]. It is a well known fact that the legal costs in the UK can be extremely expensive. A case involving serious injury or fatality caused by system failure could incur major expenditure for both compensation or legal costs. The risk associated with these unnecessary costs can be minimised or even eliminated by companies simply applying condition monitoring systems in their factories. The possible benefits to be gained from adopting condition monitoring with diagnosis systems are listed as follows.

- reduced capital investment due to less duplicated or standby machinery and fewer spare parts;
- reduced operational cost due to fewer shutdowns and increased availability leading to increased productivity;
- reduced maintenance cost due to fewer unexpected maintenance breakdowns and less consequential damage;
- reduced legal or compensatory cost due to fewer injuries and fatal accidents;

1.2 An overview of the process of condition monitoring

A complete condition monitoring process should include:

- observing, processing, and analysing the signals from different sensing instruments
- comparing these actual or processed signals with desired signals to detect malfunctions
- identifying faulty components or locations

The meaning and the process of condition monitoring can be explained using a simple example. Consider the case when you feel unwell. First of all, you could decide, according to your experience, whether or not to see a doctor. Suppose that you go to see a doctor, then what does the doctor do? Before the doctor can assess health condition, he/she must perform some routine procedures, such as asking you about your symptoms, measuring your body temperature, checking your heart rate or asking you to do some tests. Once this has been undertaken, the doctor can decide on the appropriate treatment for your problem. In some cases, you could be transferred to a hospital for further examinations using better or more advanced techniques. This example shows that health diagnosis makes use of indicating instruments including thermometers, stethoscopes and even your feelings to detect an abnormal condition. Actual signals, such as temperature readings and blood test data, are compared with the normal signals to indicate if your health condition is normal or abnormal. Finally, according to the symptoms as well as personal experiences, hopefully, the doctor can diagnoses the sickness and decide on the appropriate course of treatment for it.

The diagnostic process in engineering systems is basically the same as the health diagnosis process outline above. To diagnose an engineering system, it is necessary to collect, process, and analyse sensed data, then make use of a pertinent technique to judge if there is a fault. If a fault is detected, a decision must then be made to identify the fault according to the corresponding relation between the faulty signal and the fault. Nowadays, there are a number of techniques available to engineers for diagnosing engineering systems[5,6,7]. Some of the most popular techniques, employed in hydraulic fluid power systems, are artificial intelligence, acoustic and vibration monitoring, lubrication and contaminant monitoring, and non-destructive testing.

There are several essential points which need to be clarified further. Firstly, detecting instruments and techniques are not usually universal, in order to monitor different defects different instruments, and techniques are often required. Generally speaking, the more complex a system is, the more delicate and advanced monitoring instruments and techniques need to be. The easiest monitoring technique is the one-to-one method, one sensor to detect one defect. This implies that if a faulty signal is detected by a sensor then the fault is immediately known. However, for a complex system it is impossible to use the one-to-one method, because of the number and cost of sensors needed. Besides, the overall reliability of the monitoring system is not necessarily improved with the increased number of sensors. For this reason, it is of interest to use the minimum number of sensors to detect the maximum corresponding faults in a monitoring system. Secondly, the actual location of an indicating instrument is important. A suitable or optimal location of attachment of the sensing device must be decided before data is acquired, otherwise the resolution of the signals might not be good enough for monitoring purposes. Also, it is possible that noisy signals will disturb the diagnosis or even result in incorrect judgement. Thirdly, the data acquired from the sensors do indicate directly that a fault has been detected, and a detective method must be used to indicate an abnormal condition has occurred. The simplest technique is setting limits for individual sensor signals, if the signal is higher or lower than the limits then a fault is detected. This technique uses the measurable signals in a system as indicators for fault detection. Other indicators can also be used, such as non-measurable state variables, non-measurable process parameters, and non-measurable characteristic quantities[8]. Finally, being able to detect a faulty signal and trying to single out an unknown fault in a system are different stories. Identifying a fault in a complex system is indeed a real challenge to engineers and researchers. To complete this task, the relationships between certain faults and the corresponding faulty signals or the signatures associated with the faults must be known in advance and are provided by previous experiences and theoretical and/or experimental results. Although, the experiences of an expert are important to the success of the decision making process for identifying faults, they are not critical to it. If a system being monitored is a complicated or newly developed one, then it is likely that the expert will have limited experience and will not be capable of identifying all possible faults. Accordingly, an automatic decision making system for identifying the faults is needed for use in modern industries. Hence, there are significant benefits to be gained from using artificial intelligence related techniques, such as fuzzy reasoning[9], artificial neural networks, and expert system, which combines the experts' experiences and reasoning methods, to identify the faults. Many of these newly developed monitoring techniques will be reviewed in the following section.

1.3 Review of literature relating to condition monitoring

Because of the diversity and overlapping of the techniques proposed in the literature for monitoring and faulty diagnosis it is hard to categorise them into distinctive groups. Besides, there has been an interest in this field for a long time and the quantity of publications related to this interdisciplinary subject is so abundant that an exhaustive review of the literature is a major task. Therefore, this review only covers those the general area and basic techniques pertaining to this thesis. For distinction and convenience, the review is divided into three parts. These are general techniques, techniques having been applied to hydraulic systems, and artificial neural networks related techniques.

1.3.1 General techniques for condition monitoring

♦ **monitoring methods**

To fulfil the task of condition monitoring, an interdisciplinary knowledge is needed. Collacottt[5] focuses on mechanical fault diagnosis, and his book can be divided into three main parts. The first part deals with failure mode analysis in which different failure modes and their causes are classified and analysed. The middle part covers fault detection sensors and techniques. Special emphasis is put on vibration and oil contamination related detective techniques. These techniques are particular suitable for the monitoring of hydraulic systems. Performance trend monitoring and non-destructive testing are also considered. The final part of the book discusses fault analysis management which provides a guide for interpreting sensed data and system estimation using statistical theories. Another book for fault diagnosis and monitoring, written by Pau[6], applies statistical techniques and control theory for various purposes, including data analysis, condition and performance evaluation and fault decision. A pattern recognition approach for failure diagnosis is also discussed. Methods for processing and compressing learning pattern data is reviewed first, before classification methods used in pattern recognition are discussed, and finally an integrated automatic diagnosis system is outlined using the nearest neighbour classification rule. Traditional techniques are used in the pattern recognition and decision processes, for instance distance measure, correspondence, principal components and correlation analysis for pattern processing and Bayesian decision rule and similarity measures for class decision making. Two examples, based on the diagnosis of overall ship conditions and accelerated life testing, are given to illustrate the application of the automated diagnosis techniques.

♦ **model based monitoring**

If non-measurable system state variables are used to indicate faults, attempts can be made to estimate these state variables from measurable signals using a known system model. A comprehensive survey about decision making techniques based on the detection of abrupt changes appearing in the states and output variables of dynamic systems has been conducted by Willsky[10]. If changes in the system or in the sensors are assumed to be delta impulse or step functions, then the changes can be detected using Kalman-Bucy filters and several approaches can be used for fault decisions. The decision techniques discussed in this paper include using 'fault sensitive' filters, voting schemes, multiple hypothesis filter-detectors, jump process formulations, and innovations-based detection systems. Some of these decision techniques are briefly outlined below.

The basic idea of fault sensitive filter techniques is to design fault sensitive filters instead of optimal estimators so that particular fault modes manifest themselves as residuals in a fixed direction or in a fixed plane. Voting schemes are often useful in systems that possess a high degree of parallel hardware redundancy. For example, in standard voting schemes, there are at least three identical instruments. Simple logic is then developed to detect failures and eliminate faulty instruments. A large class of failure detection techniques involves the use of a bank of linear filters based on different hypotheses concerning the underlying system behaviour. One of the many ways that the techniques can be implemented is to hypothesise several different sets of system matrices. Filters for each of the models are constructed, and the innovations from the various filters are used to compute the conditional probability that each system model is the correct one. An abrupt change in the probabilities can be used to detect changes in real system behaviour. An innovations-based detection systems has been proposed by Mehra and Peschon[11]. They suggest that different changes in a dynamic system, such as the changes in level of input noise, the structure of the system, bias errors in instruments, etc., make the standardised innovations depart from their zero mean, unit variance and whiteness properties. Therefore, it is useful to perform statistical tests to indicate faults. These include test of whiteness, test of mean and of covariance, for example if the innovations have non zero mean indicating the possibility of bias in the instrument. According to the generalised likelihood ratio test technique, different assumed fault modes are modelled and the 'failure signature matrix' are pre-computed, which provides an explicit description of how various faults propagate through the system and filter. If the residuals are given, then the general likelihood ratio test can be carried out. The value of test ratio is used for detecting faults. Bassevlille[12] also reviewed some statistical methods for detecting changes in signals and systems but emphasised on the detection instead of the decision making. Isermann[8] in his paper surveyed a number of fault detection methods based on modelling and estimations. Following a brief summary of several basic fault detection methods, the paper concentrated on the suitable parameter estimation methods for continuous-time models. The techniques can be implemented by following the steps:

(a) Set up the equations relating the input and output variables.

(b) Establish relationship between the model parameters and the system physical coefficients are established.

(c) Use the measurements of the output and input signals to estimate the model parameters.

(d) Calculate the physical system coefficients and their changes.

(e) Refer to a catalogue of faults in which the relationship between the faults and the changes in the physical coefficients have been recorded.

Two examples are considered, the fault detection of an electrically driven centrifugal pump by parameter monitoring and the leak detection for pipelines using a special correlation method.

♦ **expert systems for monitoring**

Expert systems are software programs which use knowledge and reasoning to perform complex tasks. An expert system is essentially composed of a knowledge base containing heuristic knowledge based on human experience and expressed as rules and facts, and an inference engine which consists of reasoning or problem solving strategies for decision making. Recently they have been utilised for the purposes of condition monitoring. The reason for their use is that the diagnosis processes are knowledge intensive and experience based tasks. The application of expert systems for diagnosis widely spread over the years to a number of different areas including medical diagnosis, engines, computers, manufacturing, avionics, and communication. Pau[13] lists a number of these systems. Billington[14] built an expert system for vibration based diagnosis under the Trolex software shell. To build the expert system, firstly the normal vibration spectra, that is signatures or finger prints, are obtained from the machine using transducers applied at appropriate locations. In the expert system, this information is pre-set and macros are used to drive the data acquisition hardware. Then all of the spectral characteristics found for mechanical, electrical or aerodynamic components of the machine are correlated and the significant peaks are saved as a list for further interpretation. The next step is to define the low and high frequency limits and a threshold value, associated with the normal condition, for the individual machine elements. Status rules which are employed for confirming the correlations and explaining the cross relationships among the input data and data pertaining to normal conditions, and diagnosis rules which are used for assessing the potential faults are established. The vibration signature of an ac electric motor is used for explaining the procedures. Isermann and Freyermuth[15] constructed an on-line engineering expert system in which an analytic problem solution, a system knowledge base, a knowledge acquisition component and an inference mechanism are included. This paper has two parts. The first part presents the structure of the expert system and the function and the theory of each composing element in the structure. The second part shows experimental results obtained from applications. The technique of the analytic problem solution made use of the model based parameter estimation methods mentioned[8] and this part of the information was called the analytical knowledge in the knowledge base. Solution proposals for problems can be proposed by using analytical knowledge and co-operating with another part of the knowledge base called heuristic knowledge, which is composed of rules acquired from experience or inspection. The components in the system knowledge base consist of an analytical system model, estimation techniques for parameters and state variables, normal system behaviour, quantifiable system history, quantifiable fault statistics, which are components of analytical knowledge, fault trees, qualitative system history and qualitative fault statistics, which are components of heuristic knowledge. The inference mechanism receives symptoms generated by the analytic problem solution and a fault detection is based on fault trees. All possible faults are listed for further screening. Using the system history and fault statistics, allows the identity of the real fault to be singled out from the list of likely faults.

- **NDT methods for monitoring**

The history of using non-destructive testing (NDT) for off-line monitoring the health condition of engineering materials and structures is very long and the techniques used have attracted considerable attention recently. The reason for this is that the newly developed computerised on-line NDT monitoring systems can be easily found in the market. Scruby and Colbrook[16] describe several NDT techniques including ultrasonic, acoustic emission, magnetic related techniques and positron annihilation. The ultrasonic technique was used on-line to monitor a welding process and various defects were detected, e.g. slag, porosity, incomplete penetration etc., by letting the ultrasonic compression waves transmit through the molten metal of the weld-pool. The acoustic emission technique detects the high frequency sound generated by the monitored object using a piezoelectric transducer. Acoustic emission techniques are frequently used for monitoring cutting tool conditions, such as wear or breakage in the manufacturing processes[17]. Other usage such as monitoring bearing and pump conditions[18,19] are also very common. The magnetic properties of some materials such as steel are sensitive to physical and micro structural changes and the changes in the magnetic properties can be sensed by many techniques. The major applications are in manufacturing processes for evaluating stress levels or the depth of the harden layer in materials. The magnetic technique can also be used in contamination monitoring of oil used in fluid power system but in this case the technique is based on a different principle[20].

1.3.2 Review of literature relating to condition monitoring techniques used in hydraulic power systems

- **overview of monitoring methods for hydraulic power systems**

Hunt[7] discusses seven basic types of condition monitoring applicable to the field of fluid power systems. These are vibration, temperature, pressure, flow and leakage, contamination and fluid condition, power consumption and other miscellaneous techniques including torque, speed, displacement, and fluid level. Vibration observations seem to be the most popular and prospective monitoring technique for hydraulic power systems, especially for monitoring pumps. However, the signatures of the various fault types found in hydraulic power systems are not easily captured and it is difficult to correlate the fault modes with their associated signatures. Without these correlations it is not possible to identify the cause of the faults which have been found. Many early applications reviewed in the paper also demonstrated this dilemma. Hunt also showed several examples concerning the use of measuring fluid temperature differences for monitoring pump efficiency. Similar methods have been applied to the monitoring of valves and filters. Pressure monitoring is usually not for a particular component in the hydraulic power system but rather for the whole system. Pressure loss could mean that there is a fault caused by the pump or the control system, a fracture in the pipelines, or a loss of the load. The pressure difference upstream and downstream of a filter has been used to monitor the contamination level in oil can be found in references[20,21]. Flow and leakage are very common parameters for condition monitoring. Detected leakage flow in pumps, motors, pipelines, valves, cylinders, filters, accumulators and reservoirs can be considered as the symptoms of faults in these elements. Thermography can indicate hotspots in a hydraulic system, such as temperature rising at a point due to leakage. A technique called 'leakage flow ratio calculation'[21] was reported to be the clearest method for checking the wear condition in a particular type of pump. Nevertheless, using a flow meter could be the easiest and most accurate way to indicate the leakage in a system. The conclusion is that a low cost computerised monitoring systems with appropriate sensors is most desirable for preventing catastrophic failures and reducing life cycle costs.

♦ **contamination monitoring**

Contamination is the main cause of failure in hydraulic power systems. Blockage and abrasive or erosive wear caused by debris in the fluid, or bad fluid conditions, can give rise to system damage or deterioration in performance. Therefore, the assessment and prevention of contamination in the fluids, used in hydraulic power systems has become one of the most important and popular subjects for researchers. Five different types of contamination measurement techniques and the design of an inexpensive on-line hydraulic oil monitor have been outlined by Hunt et al[20]. This paper can be summarised as follows:

(a) Particulate sizing and counting contamination control measurement are more normally associated with laboratories because of the precision of the instrumentation. Particulate sizing can be based on visual examination using an optical microscope or automatic computerised system with optical or electron microscope or photographs. Automatic counters are available for particle counting.

(b) Particulate level assessment is a good general indication of the presence of particulate. The techniques mentioned in (a) can be used for this purpose or a membrane filter can be used to trap the contaminant for assessment. Chip plugs which detect when a certain amount of contaminant has been attracted by a magnetic field are also used. Ferrography technique is another good choice for assessing contamination level. In this case the magnetic field causes the particulate to form a substrate and by an optical density scan the relative amount of contaminant can be assessed. This method was also used in references [22-24] as will be seen later in this section.

(c) Particulate identification relates to instrumentation which describes what type of contaminant is actually present in the oil. Various techniques are available ranging from inexpensive optical microscope observation to expensive and sophisticated spectrometric oil analysis techniques or an energy dispersive x-ray analyser attached to the scanning electron microscope.

(d) A water content test is another very important item for contamination control. Chemical reaction, Dean & Stark distillation, density change, dielectric constant change, infrared absorption, Karl Fischer titration and visual inspection are possible techniques for this purpose.

(e) Fluid condition and degradation examination is concerned with the breakdown of the oil itself, which can cause system problems. Appearance, odour and viscosity are three items to be examined. The appearance and odour are usually evaluated relying on personal experience, but for viscosity measurement standard methods are available. The inexpensive and on-line oil contamination monitor designed by the authors is based on the principle of pressure drop between the two sides of a blocked filter under known flow rate is used to assess contamination levels. The specially designed oil monitor gives similar experimental results to an optical automatic particle counter but with several significant advantages at a much reduced cost. An on-line debris monitoring technique based on the above concept is described by Raw and Hunt[21], and the examples presented illustrated the value of applying it for debris monitoring. Prakash and Gandhi[22] used a direct reader ferrograph to assess the relative contamination level of an automotive vehicle hydraulic system in order to check the adequacy of the system filtration. They defined the contamination in three types, namely easy to dislodge, difficult to dislodge and impossible to dislodge. Easy to dislodge particles are the light weight loose particles easily released from the surfaces and are the major contributory factor for higher contamination levels. Those particles released to the system gradually over a long running history are classified as the difficult to dislodge type and impossible to dislodge particles are those which cannot be removed by the flow of hydraulic oil. The collected oil samples were run on a direct reader ferrograph to measure the number of large and small particles deposited in glass tube. The experimental results show that the direct reader measurements and its indices are capable of identifying the easy to dislodge and difficult to dislodge type particles and also indicate that the contamination is attributed to inadequacies in the system flushing, improper dispensing techniques and maintenance services. Therefore, combining the measurements with operating time and associated factors such as flushing, scheduled servicing and so on, it is possible to determine the contamination sources causing the system malfunctions. The technique of using ferrograhp for oil analysis also had been reported by Anderson[23] and McCullagh[24]. Recently, Hunt[25] discussed the importance of oil debris monitoring and outlined the present and future trend of research in this area.

♦ **vibration monitoring**

Backe and Schwarz[26] described a failure diagnosis system for hydraulic pumps based on vibration analysis. The vibration signals from a piezoelectric accelerometer are processed to generate RMS values and frequency spectrum. The RMS values are used to monitor the operation condition. If the RMS value from the monitored unit is over the pre-set limits for normal RMS values, then a defect is detected. The failure diagnosis uses the frequency spectrum to estimate the type and the extent of the defect. An example of the detection of a defective piston pump shows that the spectrum comparison techniques gives a different spectrum plot allowing the defect to be pinpointed. In an another example, a vane pump having a similar defect also displays a defective symptom. The authors concluded that this integrated vibration evaluation scheme could be used for qualifying condition with high reliability. The paper by Darling et al[19] demonstrated the use of acoustic emission measurement in the monitoring of high pressure hydraulic pumps. An axial piston pump was tested using acoustic emission and accelerometer measurements and the results were qualitatively compared showing that the acoustic emission technique is superior to the accelerometer technique. An earlier paper published by Bagnoli, Capitani and Citti [18] also compared the performance between acoustic emission and accelerometer measurements for monitoring bearing damage.

♦ **expert systems**

Chen[27] has developed an expert system to diagnose faults associated with electro-hydraulic servo valves based on performance tests that were carried out on three different test circuits, namely a no-load flow rate test, a pressure gain and internal leakage test, and a dynamic performance test. Using the timing when the faults occur, the author categorised the faults into three types; (i) discovered during the operating period, (ii) detected during the performance test after having been used for a certain period, (iii) found in the performance test after servicing. The rules and facts for expressing heuristic knowledge are built under the commercial expert system shell, 'personal consultant plus'. This off-line diagnosis system uses sixty-three rules generated from thirty fault symptoms, which are collected from test data and previous experience in trouble-shooting, and their combinations for reasoning. The rules and examples of using the system are included in the dissertation. Watton and Creber[28] discuss a software package for leakage flow detection in a hydraulic power system. The monitoring system used on-line data acquisition to calculate the normal values of the flow rates under current operating conditions. The structure of this system is different from the previous one which was based on an established commercial expert system shell. The system uses a mathematical model and a general program, and, without a decision making rule base, the decisions are made by applying simple two-value logic reasoning for detecting four types of leakage flow. The system was tested by introducing three cases of faults one by one in the hydraulic system and the test results seemed to be very good. An expert system, under the RuleMaster shell environment, for on-line monitoring the health condition of a hydrostatic transmission system was reported by Wang[29,30] which possessed the following features.

- capability of on-line data acquisition and system modelling.
- ability to have graphic presentation of the circuit with its current status and fault locations.
- explanations of the route of reasoning to the conclusion are available.

This system makes use of thirteen sensing devices for monitoring vibration, temperature, displacement, speed, pressure, flow rate, torque and fluid condition. These on-line acquired signals are processed and compared with the normal values that are calculated using a mathematical model of the system. The monitoring strategy is simply to set alarm reference limits to be 10% higher or lower than the normal values, except for the fluid condition monitor which uses a criterion associated with the instrument and the British Standard Contamination Code. The diagnostic strategy is embodied in the knowledge-base and expressed in If-Then rules that are arranged in terms of the fault tree analysis approach. An example is presented to show the diagnosis processes. An automated fault diagnosis expert system for hydraulic systems was designed by Atkin et al and Hogan et al[31,32]. It is a very powerful off-line fault diagnosis system and much better than Chen's system[27]. The system used failure mode and effects analysis (FMEA) and fault tree analysis (FTA) techniques for automatically assessing the integrity of a hydraulic system. This system is capable of making use of its 'component' library, which contains various hydraulic components, to build up the particular circuit under analysis. The knowledge base for fault diagnosis is built up by using deep knowledge, which is the knowledge available at the design stage, e.g. the structure or the functions of the circuit, instead of shallow knowledge, which is the knowledge represented by heuristic rules. Altogether the fault analysis method involves

- identifying all possible failure modes for all possible components in a system;
- propagating the effects of individual component failure to neighbouring components up to the system level;
- assessing the criticality of each component on the system operation.

Examples of a regenerative pump test system and a hydrostatic transmission rig were tested using this diagnosis software and the test results showed that all faults could be correctly predicted.

1.3.3 Review of literature related to the use of artificial neural networks for condition monitoring

Artificial neural network is one of the most attractive and the most active research subject, and the interest in this area is still growing. Although the field of applications of the theories of artificial neural networks in engineering is extremely broad, the application to fluid power systems is not so popular and this is especially true for condition monitoring. An effort has been made in the following literature review to cover a variety of applications using different neural networks in different research areas.

◆ **hydraulic power system monitoring**

Recently, Lu et al[33] suggested that an artificial neural network can be used for monitoring hydraulic pumps to detect and locate wear in the pump cylinders. The 'wobble' behaviour, which gives rise to pressure ripples at the pump inlet and outlet, caused by wear in the pump pistons was considered as the symptom of the piston fault. The authors made use of the signal features associated with the symptoms from sensors to train the artificial neural networks to recognise the different faults. Piston wear was simulated by introducing bigger clearance between the piston and the cylinder and different clearance sizes were related to different degrees of wear. Experimental results showed that the trained artificial neural networks were capable of correctly recognising the most faults but the success rates were not consistent. Daley and Wang[34] proposed a method to use artificial neural networks to monitor a simulated electro-hydraulic rotary drive system. In this paper, artificial neural networks were used for modelling the healthy hydraulic system and then faults can be found by comparing the actual system outputs with the neural network outputs. In order to identify faults, the signatures of parameter changes in the healthy systems are established for different fault types. Later, a specific fault can be diagnosed by evaluating the parameter changes and comparing with the known fault signatures.

- **weldment fault classification**

Baker and Windsor[35] proposed a method using the Hopfield neural network to classify the defects within steel welds. The ultrasonic data relating to four categories of defects, briefly the smooth crack, the rough crack, the slag and the pore, associated with steel welds were collected, and used to train and test the network. The ultrasonic features employed in this paper were the Kurstosis, amplitude, sphericity, pulse duration, root mean square value of amplitude with angle, and the deviation from the best-fitting plane. The data were transformed into binary images to be used in the Hopfield network for encoding and testing. The experimental results showed that the success rate of classification could reach 100%, if at least 50% of the 83 sets of defect data were used for training the network. Even if only 25% of the feature data were used for training, the accuracy of classification could still reach 95.2%. Song and Schmerr[36] reported a method of using probabilistic neural networks for flaw classification in weldments. This method also made use of the ultrasonic signatures from the flaws in the weldments to train the neural network. The probabilistic neural network was a feedforward network with four layers, comprising an input layer, pattern layer , summation layer and output layer, and the connecting weights of the network were pre-set according to the exemplar data. Accordingly, there was no need to adjust weights. A total of 239 ultrasonic waveforms were analysed for classifying cracks, porosity and slag inclusions. Ten time domain and four frequency domain features were extracted from the ultrasonic waveforms to form the components of the feature vectors that were used as the inputs to the network. The test results of flaw classification were reported to be as good as those obtained from commonly used statistical approaches, such as the K-nearest neighbour method.

♦ **cutting tool monitoring**

Rangwala and Dornfeld[37] suggest that the spectrum of acoustic emission and of cutting force could be used for monitoring the wear of cutting tools. A feature selection method was applied to 30 samples. Three sets of features were used for testing a feedforward multi-layer network and it was found that the outcome of the successful classification rate were different with different feature sets. The classification success rate was reported to be as high as 97%. The comparison of the performance between the multi-layer network and perceptrons showed that the former was superior to the latter. Using neural networks to monitor the condition of the cutting tools is also reported in other literature including [38,39,40].

♦ **quality control monitoring**

A method for estimating surface roughness and bore tolerance in circular end milling using the feedforward multi-layer network is described by Okafor, Marcus and Tipirneni[41]. The first part of this paper proposed a neural network structure for the trend monitoring of the machining process and the second part addressed the actual implemented neural structure and the details of the estimation method. The estimation method used two three layer networks for independently estimating the surface roughness and the bore tolerance, each of them had three input units and one output unit and a certain number of hidden units between them. The input feature vectors had three components which were the root mean squared values of the cutting forces in the x and y directions, the acoustic emission peak magnitudes, and the spindle acceleration. Tests were carried out to train as well as to test the network. Finally, in the discussion section, the authors reported that the experimental results demonstrated the feasibility of using a relatively simple neural network to integrate multiple-sensor information to obtain a fairly accurate estimate of surface roughness and bore tolerance in circular end-milling.

♦ **aircraft diagnosis**

Aylward et al[42] describes a method for the diagnosis of in-flight faults on an aircraft using neural networks. According to the paper, the main reasons for trying to employ the neural networks for fault diagnosis could be summarised as

- ground based diagnostic systems and diagnostic procedures are expensive and ineffective;
- neural networks have the capacities of modelling complex relationships and rapidly processing signals using small amount of computer memory;
- the possibility of expanding the trained networks to accommodate newly discovered faults;
- relatively low development costs compared with building an expert system.

The authors suggested a four layer back-propagation neural network for fault isolation and a three layer back-propagation network for damage detection and estimation when applied to an inertial navigation system. The test results showed that the fault isolation network could perform better than the expert system approach under noisy flight conditions. The fault detection and estimation network was proved to be superior to the mathematical model based system.

1.4 Objectives of this research project

The literature review indicates that condition monitoring systems used for monitoring hydraulic systems are mainly expert systems. The diagnostic performance of these systems relies on past experience or/and requires a large number of transducers. Artificial neural networks have been found suitable for system modelling and pattern recognition. These capabilities make it suitable for use in condition monitoring and offer new techniques which can be better in its performance than traditional monitoring systems. Artificial neural networks also have other attractive properties such as high computing rate, fault tolerance, and learning ability. However, during the period of the paper review, only two papers which applied artificial neural networks to hydraulic system condition monitoring, were found. These papers applied just one type of artificial neural network, and many other types of artificial neural networks can be used. Therefore, the application of artificial neural networks to the condition monitoring of hydraulic systems is an area suitable for further research.

Based on the above considerations, the main objectives of this research project were set to be

(a) to investigate the different methods used for condition monitoring.

(b) to study different artificial neural networks and the feasibility of using artificial neural networks in monitoring fluid power systems.

(c) to develop the simulation programs associated with the algorithms of different artificial neural networks for future applications.

(d) to demonstrate the use and compare the performance of different types of neural networks for fault classifications on two test rigs.

(e) to study the feasibility of using the techniques derived from fuzzy logic for the condition monitoring of the fluid power systems.

Chapter 2
The Basic Concepts of Artificial Neural Networks

2.1 A brief history of the development of artificial neural networks

The history of research into the human brain goes back a long time, but the earliest record of an attempt to build a mathematical model for simulating the function of human brain is attributed to McCulloch and Pitts in 1943 stated in[43,44]. According to their research, they proposed the simple model shown in Figure 2.1. Although it is indeed very simple from today's point of view, it is the foundation stone of modern artificial neural networks. Since then many advanced neuronal models have been suggested based on this simple model. The details of this model and other evolved models will be discussed in section 2.3.

In 1949, Hebb[43-45] provided an explanation of the contribution of synapses to the process of learning and the learning rule developed from his theory is called the Hebbian learning rule. The Hebbian rule states that if a neuron A persistently or repeatedly takes part in firing a neuron B, then the weight between them should be strengthened. The mathematical form of this theory will be shown in section 2.5. Rosenblatt[43,45] in 1958 introduced the first concrete network model, named perceptron, which incorporated learning into the McCulloch and Pitts model. The work of Rosenblatt has been influential in the development of modern adaptive networks. In the 1960s, the perceptron concept attracted much research. Rosenblatt[43] himself also presented a learning rule for weight adaptation, which was called the back-coupled error correction. In this rule, the error was defined as the difference between the desired output and the actual output. According to the learning algorithm, the amount of adjustment of each weight was proportional to the error. The model of the elementary perceptron, also called the one-layer perceptron, is shown in

Figure 2.2. Although the perceptrons can handle certain classes of problems well, there are some serious limitations in their application, including poor generalisation, lengthy supervised off-line learning, and an inability to non-linearly separate pattern classes[46].

In 1960 the Adaline (adaptive linear element) shown in the Figure 2.3, was developed by Widrow and Hoff[47]. It is also a neuronal model with learning capability. This model employs a learning rule for weight adjusting which is similar to the one used in the elementary perceptron, the major difference being that the error in this model is defined as the difference between the desired and the actual values of summed weighted inputs. The learning algorithm was known as the least mean square(LMS) error correction rule or Widrow-Hoff delta rule. Later, an extension of the Adaline model was reported, called Madaline which means many Adalines. The basic Madaline is a network which has many Adalines arranged in a three-layer (including the input layer) feedforward structure. Adaline and Madaline have been successfully applied in many engineering areas, such as control, pattern recognition and communications.

The growth of research into artificial neural networks was stunted for a period, from 1969 to 1981, because of a serious criticism about the limitation of the perceptron made by Minsky and Papert[48]. But there were some influential researchers, such as Anderson, Grossberg, Hopfield, Kohonen, Kosko, still working in this field and their work finally resulted in many important findings, such as Brain-State-in-a-box by Anderson[45], neocognition by Fukushima[49], and adaptive resonance theory by Grossberg[50-52]. The adaptive resonance theory finally led to the advent of three versions of the ART networks, briefly ART 1, 2 and 3. All of the ART networks are able to achieve stable self-organisation of recognition codes for arbitrary sequences of input patterns. The ART 1 accepts only binary input patterns, but the ART 2 can be used for either binary or analogue input patterns. In the newest version ART 3, Carpenter and Grossberg ingeniously infused the concepts of transmitter dynamics occurring in biological neurons into the original ART architecture. However, unlike the famous ART 1 and 2, it seems that this new model has not yet drawn much attention.

A neural network model called the Hopfield network was presented in 1982[53] which was a fully interconnected network that could only accept binary or bipolar data and applied the basic Hebbian rule for training. The Hopfield model was said to be a type of network with content-addressable memory. A content-addressable memory network is capable of giving rise to correct output by inputting a pattern which is only a sufficient part of the original pattern. The Hopfield network has two major limitations in its application of the content addressable memory[46]. The first one pertains to the ability of pattern storage when used as the content addressable memory. The number of classes of patterns which can be kept in the network is seriously limited to less than 0.15 times the number of processing units in the network. The second limitation is that a training pattern could generate an incorrect output if the pattern shares many bits in common with another pattern. Hopfield[54] also proposed another version of Hopfield model in 1984, which could handle analogue data. The Hopfield networks are one of the most successful networks that have been extensively applied in numerous areas, including image processing, control, signal processing, and pattern recognition[45].

The human brain reacts to the surroundings with the aid of the senses and on the brain surface maps exist which are associated with certain parts of human body. The self-organising map of Kohonen[55] was constructed based on this biological evidence. The network is a two-layer network with its output layer of processing units arranged in a two dimensional grid. The units in the output layer are specifically tuned to correctly respond to the various input classes of patterns through a self-tuning or unsupervised learning process. The clusters of classes of exemplar inputs can be automatically formed on the two dimensional map. This network can also employ a supervised learning scheme to train it as a pattern classifier. The new training algorithm was called the learning vector quantization(LVQ) method. A phonetic typewriter[56] has been tested using this network. The results showed that the correctness of any letter was as high as 92 to 97% and the phonetic typewriter has been implemented in several hardware versions.

A probabilistic model of a neural network was reported by Ackley, Hinton and Sejnowski in 1985[57]. This model, called the Boltzmann machine, is a Hopfield network that settled into solutions by a simulated annealing process governed by Boltzmann statistics. Because of the need for a very large calculation power for the implement of the Boltzmann machine, the real time applications were few. The counterpropagation network[58] was a type of network that made use of the structures and learning algorithms used in the Kohonen self-organising map and the Grossberg outstar neurons. This network, the author claimed, would self-organize a near-optimal lookup table approximation to the mapping used to generate its data and this method worked equally well for both binary and continuous vector mappings. After training, the network could produce the correct output even when given partially incomplete or incorrect inputs.

The most influential work in the development of artificial neural networks is probably that published by Rumelhart, Hinton and Williams[59]. The so-called error back-propagation training algorithm solved the dilemma of a perceptron which could not be trained if the layer of processing units was over two. From then on the research in artificial neural networks gained in momentum. Nowadays, back-propagation neural networks are still the most popular networks for engineering applications, especially in control engineering. The back-propagation network, the self-organising map, and the ART networks will be discussed in more detail in chapter 3. Following the steady development in the artificial neural networks during the past decade, many researchers have been involved in exploration of new theories, new training techniques, and suitable areas of application.

2.2 Biological neurons

Biological nerve cells or neurons[44,60,61] are fundamental elements of the nerve systems or networks in the human body. The most important, most exquisite and most powerful neural network in the human body is the brain in which numerous neurons are interconnected to form a huge and complex structure which performs incredibly sophisticated functions. There are various types of neurons in the human brain with different characteristics. A typical biological neuron and related structure are shown in Figures 2.4 and 2.5. The biological neuron is composed of three parts: the cell body, the dendrites, and the axon. The cell body contains the nucleus and carries out the biochemical transformations necessary to the life of the neuron. The hair-like dendrites serve as receivers for the incoming electrical pulses or signals from other neurons. The axon with its branches acts as a transmitter for the electrical pulses to other neurons. The connecting junctions among the neurons are called synapses which are the most important constituent part of the biological neurons, because, according the neurobiological researches, the activities of synapses are responsible for the learning ability and memory of the neural systems. There is no direct contact point between two abutting neurons as shown in Figure 2.5, instead the two neurons are separated by a space called a synaptic gap, which the electrical pulses can not directly cross. At the synapses the electrical pulses that are triggered by the pre-synaptic neuron are transduced into certain chemicals called neurotransmitters. These are passed through the pre-synaptic membrane and released into the synaptic gap. On the other hand, the released neurotransmitters diffuse onto the post-synaptic membrane and affect the potential difference across the membrane of the post-synaptic neuron which in turn generate the electrical pulses depending on the types of synapses involved. If the effect of the biochemical processes of a synapse on the target neuron results in restricting the potential of the generating pulses, then the synapse is of the inhibitory type. If the effect tends to enhance the capacity of pulse generation, then the synapse is of the excitatory type. Messages or signal exchange occurs as a result of the neurotransmitters releasing and absorbing processes taking place at the synaptic junctions. When the strength of the accumulated small input pulses in a neuron reaches a threshold value, a large pulse is created and transmitted via the axon to neighbouring neurons.

Figure 2.6, and the following brief summary of the processes of signal transfer among biological neurons, will be helpful in making the similarity between the biological neuron and the mathematical neuronal model clear.

(a) The cell body receives pulses transmitted from its own individual dendrites.

(b) If the total strength of the input pulses is higher than some threshold value, then an output pulse will be triggered, otherwise there is no output pulse from the neuron.

(c) When a pulse is triggered by the cell body, the axon will transmit it to the dendrites which belong to abutting neurons.

(d) When the pulses arrive at the synaptic junctions, they are transferred into chemical substances which will affect the membrane potential of target neurons to create electrical pulses.

In this way, pulses as well as messages are passed very quickly and very efficiently through a neural system.

2.3 Neuronal models

The model of an artificial neuron, which is often called a processing unit, perceptron or Adaline, is the mathematical model of the biological neuron based on the knowledge of the structure and activities of the biological neurons. The first mathematical model was proposed by McCulloch and Pitts as shown in the Figure 2.1. This model makes use of computational procedures to simulate the activities of the neurons. The basic idea of this model can be summarised as follows.

(a) All calculations are completed in discrete time intervals.

(b) Each incoming input x_i is multiplied by a constant weight w_{ij} before arriving at the processing unit.

(c) The sum of the weighted inputs, including an inhibitative input with minus sign, is compared with a threshold value h>0. If the value of the sum is higher than the threshold value then the processing unit is activated, otherwise the unit is not activated.

(d) The processing unit sends an output to other abutting units through a binary threshold function or Heaviside function. The output from the binary activation function is 1 if the unit is activated or 0 if the unit is not activated.

(e) The weight associated with each passageway is not changed with time. This implies that there is no learning in the model.

The elementary perceptron model[46], shown in Figure 2.2, is the first neuronal model capable of learning. The structure of this model has two layers in which the output layer has only one output unit connected with a certain number of input units. The activity level of the output unit are calculated by equation 2.1 and the output from the unit is given according to equation 2.2. For simplicity and without losing generality, usually the threshold value is included in the weighted inputs. The equation for updating weights in discrete time form is described by the expression given the equation 2.3.

$$s(t) = \sum_{i=0}^{n} w_i(t)x_i(t) \tag{2.1}$$

$$o(t) = f(s(t)) \tag{2.2}$$

$$w_i(t+1) = w_i(t) + \mu[d(t) - o(t)]x_i(t) \tag{2.3}$$

where x_0=1 and $x_i(t)$ is the ith component of the input vector. $w_0(t)$ represents the threshold value. s(t) and o(t) are the sum of the weighted inputs and output of the output processing unit. f(*) is the binary activation function so that o=1 if s>0 and o=0 if s≤0, but bipolar activation function is also used in the literature. $w_i(t+1)$ is the weight connecting the ith input and the output processing unit at time t+1 and $w_i(t)$ is the same weight at time t. μ is the learning rate. d(t) is the desired value of the output.

Another simple neuronal model whose basic concept is similar to the elementary perceptron is the Adaline model, shown in Figure 2.3. The structure of the Adaline can be separated into two parts. The first part is called the adaptive linear combiner(ALC) and the second part is the bipolar output function. The major difference between the previous model and the current one, as mentioned before, is the definition of the error used in the learning rule. As shown in the Figure 2.3, the error is obtained by subtracting the output value of the ALC from a desired value d and then this error is fed back to adjust the weights. The governing equations of Adaline are given in equation 2.4-2.6.

The equation of summation $$s(t) = \sum_{i=0}^{n} w_i(t)x_i(t) \tag{2.4}$$

The equation of output $$o(t) = g(s(t)) \tag{2.5}$$

The equation for adjusting weights

$$w_i(t+1) = w_i(t) + \mu[d(t) - s(t)]x_i(t) \tag{2.6}$$

where g(*) is the bipolar activation function so that o=1 if s>0 and o=-1 if s≤0. d(t) is the desired value of the ALC output.

A general model without learning ability is illustrated in references[62,63] and shown in Figure 2.7. The equations for describing this model in the continuous time domain are listed as follows.

(a) The equation of summation

$$s_j(t) = \sum_{i=0}^{n} w_{ij}x_i(t) + \sum_{r=1}^{m} v_{rj}o_r(t) \tag{2.7}$$

where $s_j(t)$ represents the sum of the total weighted inputs to the jth processing unit at time t.

w_{ij} is the weight of jth processing unit associated with the ith external inputs $x_i(t)$ and v_{rj} is the weight of jth processing unit associated with the rth feedback inputs $o_r(t)$.

n and m are the total number of external inputs and feedback inputs respectively.

(b) The equation for the dynamic system, shown in Figure 2.7, can take the form as either

$$c_0 \dot{a}_j(t) + c_1 a_j(t) = s_j(t) \tag{2.8}$$

or

$$a_j(t) = s_j(t-\tau) \tag{2.9}$$

where $a_j(t)$ gives the activity of the linear dynamic system at time t and $\dot{a}_j(t)$ is the time derivative of $a_j(t)$. c_0, c_1 and τ are constants.

Different variants of the dynamic system can be derived by setting appropriate values to the constants c_0 and c_1 in equation 2.8. For example, if c_0 is set to be zero and c_1 is equal to 1, then the system becomes non-dynamic, i.e.

$$a_j(t) = s_j(t) \tag{2.10}$$

(c) The output-input relation for the non-dynamic activation function can be expressed as

$$o_j(t) = f(a_j(t)) \tag{2.11}$$

where $o_j(t)$ represents the output of the jth processing unit at time t and f(*) is the non-dynamic activation function.

The equation 2.10 shows that the weighted sum of inputs to a processing unit is employed as the activity value of the unit. There is no transformation from the value of the sum to the value of activity. This is the case usually seen in the literature. Accordingly, equation 2.11 is often written as equation 2.12. Other examples are shown in equation 2.2 and 2.5.

$$o_j(t) = f(s_j(t)) \tag{2.12}$$

There are a large number of activation functions which can be chosen in addition to the binary and bipolar functions used in the single-layer perceptron and Adaline model. Some of the commonly employed functions, such as logistic or sigmoidal, hyperbolic tangent, saturated function, and quadratic function are shown in Figure 2.8.

There are still many other models of artificial neurons, such as Fukushima model, the linear associative memory model, Grossberg model, Hopfield model, etc. that can be found in the literature [43,44,49,53,64].

2.4 Structures of artificial neural networks

A single biological neuron without proper connections with other neurons is limited to execute simple functions. However, a well organised neural network in which a large quantity of neurons are interconnected, and each member neuron co-operates with its neighbouring neurons can carry out highly difficult tasks. For example, there is an extremely important part in the human brain called the cerebral cortex that is responsible for the information processing capability. The cortex is only about three millimetres thick and is composed of six layers of connected neurons almost everywhere[61]. The number of densely packed neurons in this area of the human brain is estimated as about 100 billion. The co-operation of these linked neurons allows the human to process the variety of information gathered from the surroundings and make decisions with speed and efficiency. For the sake of mimicking the biological neural networks and performing a particular task, the artificial neural neurons must be properly interconnected to form a specified structure which is the so-called artificial neural network.

The artificial neural network can be found in literature with other names, such as parallel distributed processing model or connectionist model. No matter what the name is, all these models can be characterised by individual neuron features(e.g. activation function), architecture or connection style, and learning algorithm[46]. All of these features will become clearer later. A variety of architectures for artificial neural networks have been suggested and most of them are constructed from some basic structures. Several basic structures of artificial neural networks, without including their hierarchical architecture, are summarised as follows.

(a) Multi-layer feedforward network. A multi-layer feedforward network, shown in Figure 2.9(a) , is built up by processing units which are arranged in different layers and the connections exist only between the units located in the different layers. Signal flow in the network of this category is restricted in the forward direction, but it is possible for the signal to feedback from the output units in the last layer to the input units in the first layer. The most popular back-propagation network is one of many networks built on this basic structure.

(b) Two layer network with feedback connections. A network of this category is shown in Figure 2.9(b) which has two layers of processing units and the links between the units include not only feedforward paths but also feedback paths. Therefore, signals can be exchanged in the forward and backward directions. The famous adaptive resonance theory networks make use of this type of structure.

(c) Single layer self-enhancing and lateral feedback networks. The network of this type, shown in Figure 2.9(c), has only one layer of processing units, but each individual unit is fully connected with every other units in the network. Besides, the processing units have a self-feedback or self-enhancing connection to themselves. Using this structure each unit in the network is capable of sending its output signal laterally to every other unit and also to itself, usually to inhibit the activity of other units and to enhance the activity of itself.

(d) Randomly connected networks. The theory of the brain being a randomly connected network, when viewed at a macroscopic level, was employed to build the artificial neural networks with random connections[66].

2.5 Learning modes and learning rules

The capability of the human brain to process information, to memorise and to deal with changes in the surroundings comes from the processes of learning. Learning in the biological neurons is due to the adaptive changes in synapses. In artificial neural networks, the key to the learning is the ability to adjust the connecting weights or the memory between processing units so that the desired outputs can be produced after the corresponding inputs are presented. Learning needs a learning algorithm. The learning algorithm for a particular artificial neural network tells the network when and how to modify its weights. The general mathematical expression for learning is given in the continuous time form in equation 2.13. The continuous time learning rule is often transformed into the discrete time form for the convenience of simulation in digital computers. The general approximate discrete time learning rule takes the form of equation 2.14. Sometimes a linear or nonlinear passive decay term is added to the continuous time equation 2.13 to simulate memory decay in the real neurons if there is a lack of external stimulation[43,65].

$$\frac{dw_{ij}(t)}{dt} = \mu(t)g(a_i(t),a_j(t),o_i(t),o_j(t),w_{ij}(t)) \neq 0 \qquad (2.13)$$

$$w_{ij}(t+1) = w_{ij}(t) + \mu(t)g(a_i(t),a_j(t),o_i(t),o_j(t),w_{ij}(t)) \neq 0 \qquad (2.14)$$

where $w_{ij}(t)$ is the connecting weight from the ith to the jth processing unit.

$a_i(t)$ gives the activity of the ith processing unit.

$\mu(t)$ is the learning rate and g(*) represents certain function relating the weights, activities and outputs from both processing units.

Hopefully, following the learning algorithm the weights of the artificial neural network can at last settle down to a stable state and the performance of the network can satisfy the requirement of a particular application.

2.5.1 Learning modes

Depending on the learning algorithms used for training artificial neural networks, the learning can be roughly divided into two main modes, namely supervised learning and unsupervised learning. During the process of supervised learning, there exists an external "teacher" monitoring the progress of the learning. The "teacher" could be certain pre-set goals by which the learning error information is generated and accordingly the connecting weights of the network are adjusted step by step. The "teacher" also decides when to terminate the learning. On the contrary, the networks undergoing the unsupervised learning process do not need any external "teacher" or influence. The progress of learning in these networks is directed by an internal performance criterion embodied in the learning algorithm. Following the instructions of the learning algorithm, the networks look for regularities and trends in the exemplar inputs to modify the connecting weights.

2.5.2 Learning rules

(a) Hebbian learning rules

The earliest learning rule is the Hebbian rule as mentioned in the section 2.2(often called the simple Hebbian learning rule). Many learning rules evolved from this simple rule, such as signal Hebbian learning, Kosko differential Hebbian learning, random differential Hebbian learning, and drive reinforcement learning[43,45,60,64]. The simple Hebbian learning rule is also called the simple Hebbian correlation rule where the value of w_{ij} is the correlation of the output o_i of ith unit and the activity level a_j of the jth unit. The equations of the continuous time form of the simple Hebbian learning rule without and with a linear passive decay or forgetting term are given in equations 2.15 and 2.16. The approximation of the continuous time learning equations in the discrete time is written as equation 2.17.

$$\frac{dw_{ij}(t)}{dt} = o_i(t)a_j(t) \qquad (2.15)$$

$$\frac{dw_{ij}(t)}{dt} = -w_{ij}(t) + o_i(t)a_j(t) \qquad (2.16)$$

$$w_{ij}(t+1) = w_{ij}(t) + o_i(t)a_j(t) \qquad (2.17)$$

Frequently, the term $o_i(t)a_j(t)$ in the above equations is taken in the variant forms, such as $s_i(t)s_j(t)$ or $a_i(t)a_j(t)$[45,57].

The outputs of an artificial neuron are frequently called signals. The signal Hebbian learning rule makes use of the correlation of the output signals of the input and output units by adjusting the weights instead of using the correlation of the output signal and the activity as in the simple Hebbian learning. Similarly, the differential Hebbian learning employs the correlation of signal velocities to modify the weights. Two versions of continuous time signal Hebbian learning rules are given in equations 2.18 and 2.19 and the discrete time learning is shown in equation 2.20.

$$\frac{dw_{ij}(t)}{dt} = f(a_i(t))f(a_j(t)) \qquad (2.18)$$

$$\frac{dw_{ij}(t)}{dt} = -w_{ij}(t) + f(a_i(t))f(a_j(t)) \qquad (2.19)$$

$$w_{ij}(t+1) = w_{ij}(t) + f(a_i(t))f(a_j(t)) \qquad (2.20)$$

where $f(a_i)$ represents the output signal of the ith processing unit.

The differential Hebbian learning rule takes one of the following forms

$$\frac{dw_{ij}(t)}{dt} = \dot{f}(a_i(t))\dot{f}(a_j(t)) \qquad (2.21)$$

alternatively $$\frac{dw_{ij}(t)}{dt} = -w_{ij}(t) + \dot{f}(a_i(t))\dot{f}(a_j(t)) \qquad (2.22)$$

discrete time form $$w_{ij}(t+1) = w_{ij}(t) + df(a_i(t))df(a_j(t)) \qquad (2.23)$$

where $\dot{f}(a_i)$ represents the time derivative of the output signal of the ith processing unit and $df(a_i)$ stands for the expression $f(a_i(t))-f(a_i(t-1))$.

(b) Error-correcting learning rules

The rules in this category are used in the supervised learning mode. The error of the calculated output from the output units are compared with the desired or target values to produce errors and then the errors are used to modify the connecting weights. The learning rules for the previously mentioned elementary perceptron and Adaline, and error back-propagation networks belong to this category.

(c) Instar and outstar learning rules

Two key components extensively used in the class of adaptive resonance theory and other networks are the instars and outstars[43]. Both are classified in the unsupervised learning mode. The instar fan-in and outstar fan-out structures are shown in Figure 2.10. A typical learning rule for the instar coding can be expressed in the continuous time form as

$$\frac{dw_{ij}(t)}{dt} = \mu(t)[o_i(t) - w_{ij}(t)]o_j(t) \qquad (2.24)$$

or alternatively in the discrete time version as

$$w_{ij}(t+1) = w_{ij}(t) + \mu(t)[o_i(t) - w_{ij}(t)]o_j(t) \qquad (2.25)$$

The correction term in equations 2.24 and 2.25, i.e. $[o_i(t)-w_{ij}(t)]o_j(t)$, shows that this rule has the nature of competitive learning. According to this rule, the weights are adjusted for only those connected to the winning unit, which is active, in the upper layer and the other weights not connected to the winning unit are left intact. Suppose that the term on the right hand side of equation 2.4 is distributed to two terms, then it is clear that this equation is similar to equation 2.16 except that a nonlinear forgetting term is used instead of a linear one. A variant of equation 2.25 is also found in literature[60]. In the later case the $o_j(t)$ in the equation 2.5 are all set to be 1, namely

$$w_{ij}(t+1) = w_{ij}(t) + \mu(t)[o_i(t) - w_{ij}(t)] \qquad (2.26)$$

On the other hand, the typical learning rules for the outstar pattern learning are given by equation 2.27 to 2.29 in the continuous and discrete time versions.

$$\frac{dw_{ji}(t)}{dt} = \mu(t)[o_i(t) - w_{ji}(t)]o_j(t) \qquad (2.27)$$

or $$w_{ji}(t+1) = w_{ji}(t) + \mu(t)[o_i(t) - w_{ji}(t)]o_j(t) \qquad (2.28)$$

or $$w_{ji}(t+1) = w_{ji}(t) + \mu(t)[o_i(t) - w_{ji}(t)] \qquad (2.29)$$

2.6 Closure

In this chapter the historical development of artificial neural networks was introduced and different architectures are explained. The learning rules covered in this section are only those for general usage without considering the architecture of the networks. The use of some of the learning rules will be shown in the next chapter. However, one should differentiate between learning rules with learning algorithms. A learning algorithm is specially designed for training a particular type of network and a learning rule is used in an algorithm for adjusting the memories in an artificial neural network. The relationship between the learning rule and the learning algorithm will become clear, after three types of artificial neural networks and associated training algorithms are introduced. The specific application of artificial neural networks to fault diagnosis in fluid power system is illustrated later in this thesis.

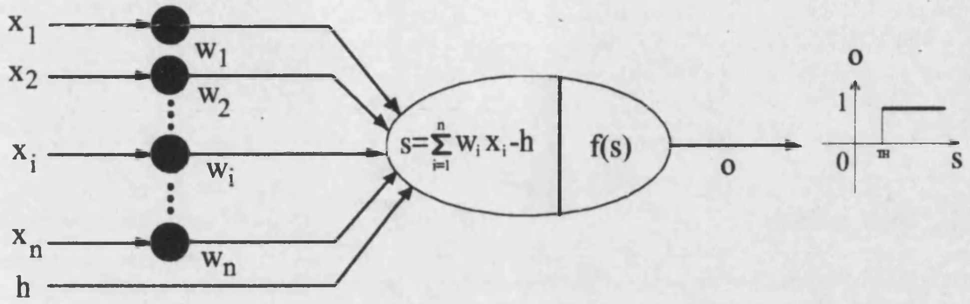


Figure 2.1 McCulloch-Pitts neuronal model

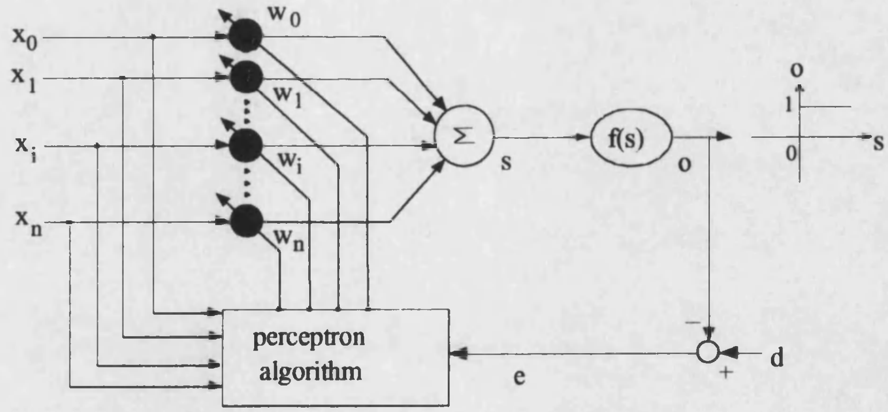


Figure 2.2 Elementary perceptron model

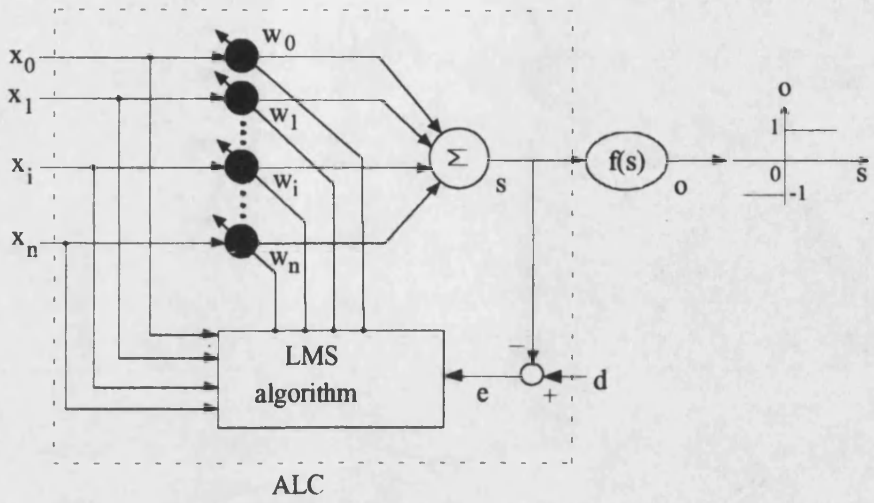


Figure 2.3 Adaline model

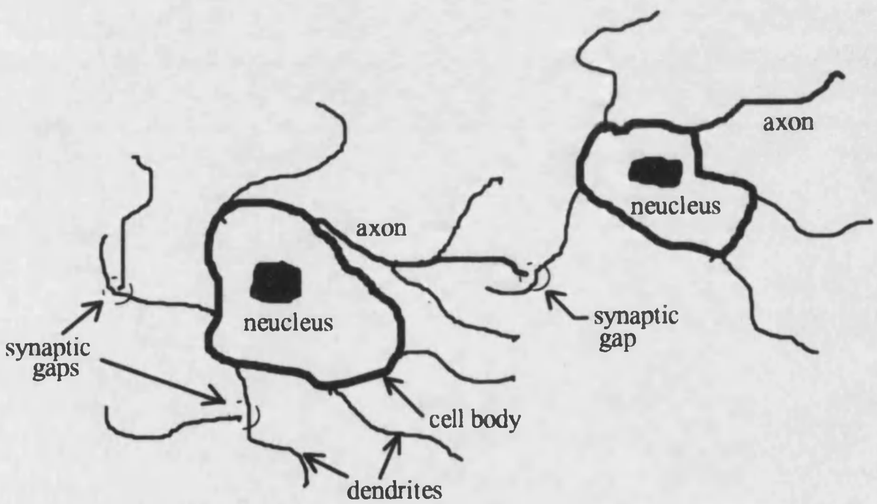


Figure 2.4 A schematic diagram of biological neurons

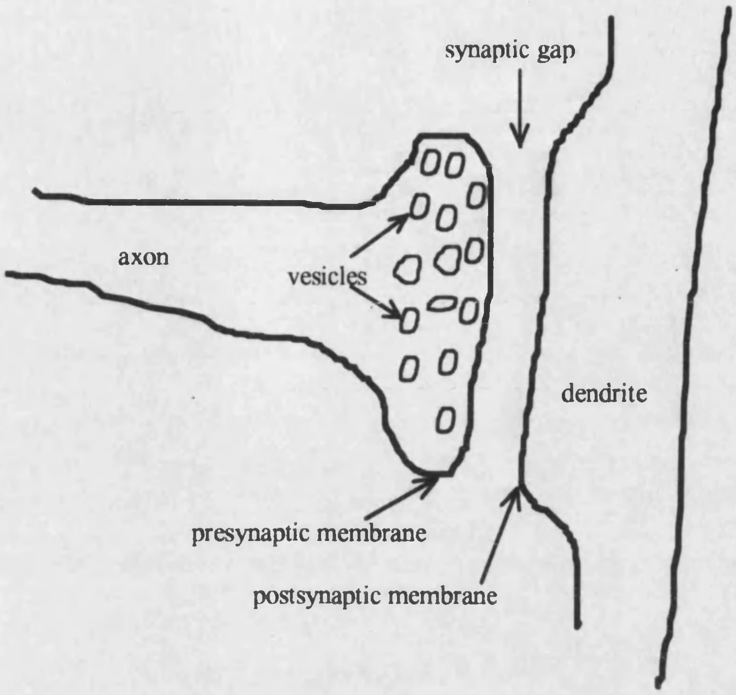


Figure 2.5 A schematic diagram of the synaptic gap

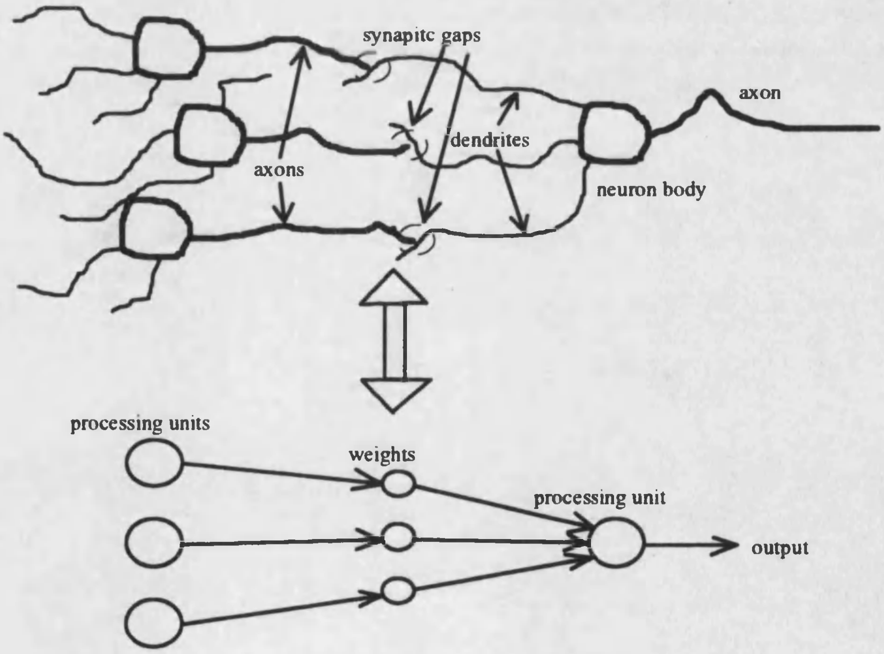


Figure 2.6 Comparison of biological neurons and the neuronal model

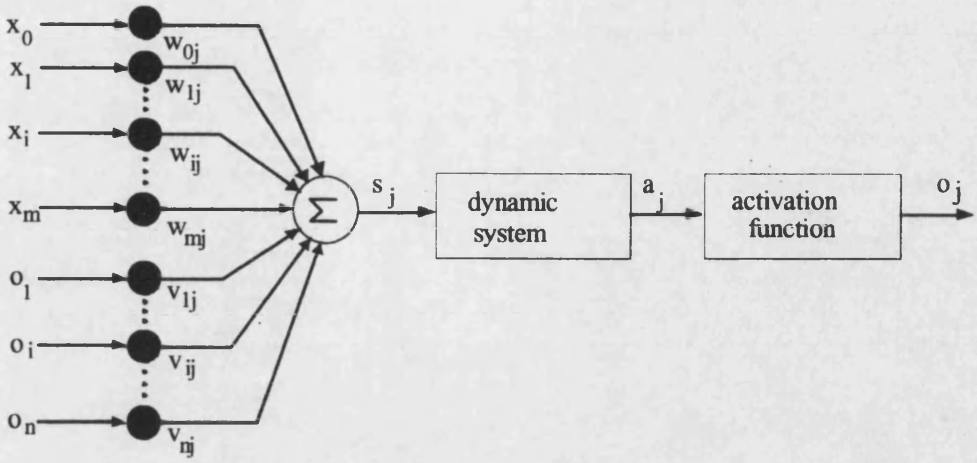


Figure 2.7 General model

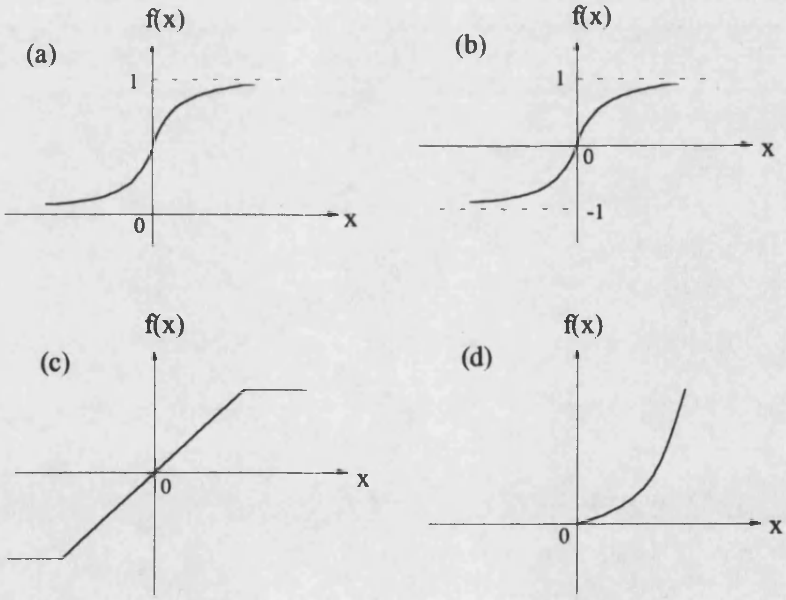


Figure 2.8 Examples of activation functions. (a) Sigmoid function. (b) Hyperbolic tangent. (c) Saturation function. (d) Quadratic function.

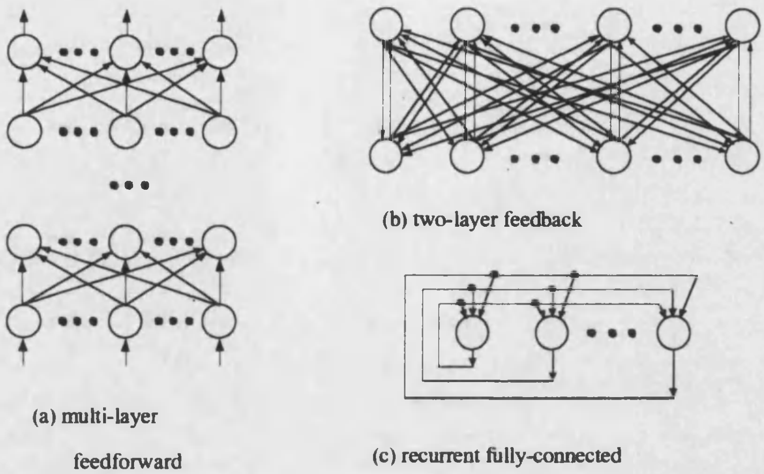


Figure 2.9 Some basic structures of artificial neural networks

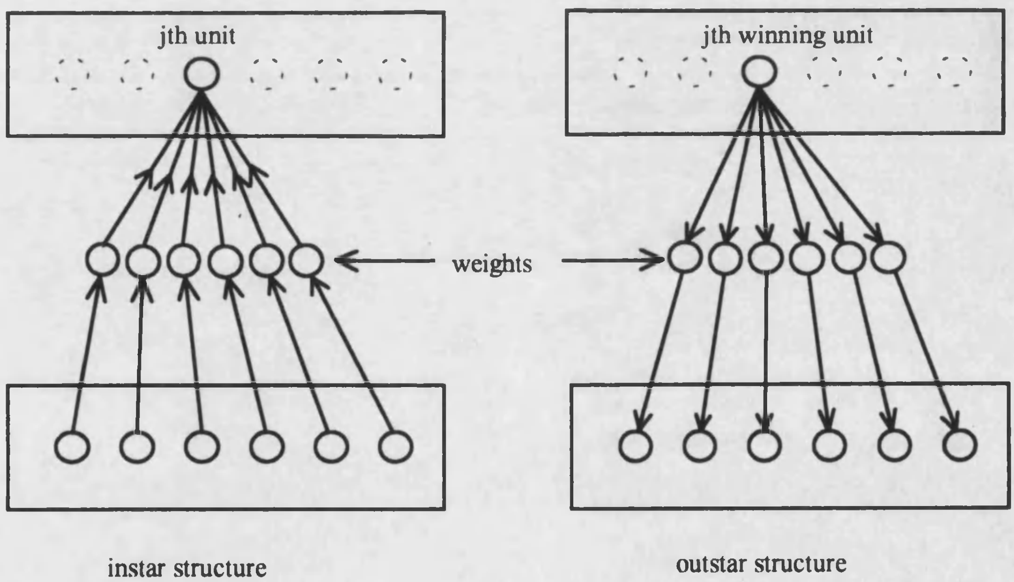


Figure 2.10 Instar and outstar structures

Chapter 3
Artificial Neural Networks

3.1 Introduction

In this chapter the basic concepts of three major groups of artificial neural networks will be reviewed. Some examples relevant to each of these groups, which were originally used for testing the programs written by the author, will be demonstrated in order to show the capacities of these networks. The artificial networks covered in this chapter include the multi-layer perceptron(multi-layer back-propagation network), the self-organising map(SOM) and the adaptive resonance theory(ART).

Artificial neural networks are mathematical models of biological neural systems, which are composed of a number of highly interconnected layers of simple neuron-like processing units in specified structures. A variety of artificial neural networks have been developed and implemented in the past[45,66,67]. Vemuri[66] listed thirteen best-known neural networks and Simpson[45] analysed twenty-eight artificial neural networks and their learning paradigms. No matter the type of neural network, the characteristics of the neural network can be specified by several major features[59]:

- the characteristics of the processing units, including the dynamics of activation and the activation or threshold function
- the connection of the processing units, namely the structure of the network
- the rule guiding the direction of signal flow
- the learning algorithm employed for the encoding process

All of these features were explained in the previous chapter except the learning algorithms related to the special neural networks. In this chapter, some learning algorithms as well as associated neural networks, which are applied in this research, will be illustrated.

The rapid growth of interest of artificial neural networks is mainly due to their unique properties indicated below[46,62,68]:

- High computing rate and fault tolerance. Neural networks usually have massive, highly connected, computing units and this structure, if implemented in hardware, will provide the ability of parallel computation. On the other hand, because of a large number of highly interconnected computing units existing in the neural networks, damage to a few units or links will not significantly affect the overall performance of the networks.
- Learning ability. Artificial neural networks can adjust their behaviour in response to a changing environment by modifying their parameters; they can learn like we do. This learning ability is essential in the area of pattern recognition, which is the main issue of many applications, such as speech and word recognition, signal recognition, fault classification.
- Capability of generalisation. A learned neural network stores information in its whole structure distributedly, not locally. Accordingly, it is usually insensitive to minor changes in the features of its inputs. For example, if an input is a part of a learned image or a noise coupled pattern, it is still possible for the original image or pattern to be successfully recalled from the learned network. Also, some of the neural networks are theoretically able to learn to approximate continuous function to arbitrary accuracy. This makes them very useful in many applications, especially in control engineering.

The areas of applications of artificial neural networks are extremely extensive and still expanding rapidly[45,68-70]. These applications can be roughly categorised as robotics and control, image and signal processing, manufacturing, automated inspection and monitoring, speech and writing recognition, medical diagnostics, business and finance, environment and agriculture, and military. However, although artificial neural network techniques seem very powerful for solving many problems, they are not panaceas. It is believed that the incorporation of artificial neural network techniques with other techniques, for example fuzzy logic, is still required[68,71,72].

3.2 Multi-layer back-propagation(BP) networks

The real meaning of multi-layer back-propagation networks should be understood as the multi-layer perceptrons with the back-propagation(BP) training algorithm. This group of networks is often called the BP network in the literature. The structure of the BP network is based on the structure of the elementary perceptron and the learning algorithm is the generalisation of the steepest descent searching technique used in the Adaline[59,73]. Since the publication of the backpropagation method by Rumelhart et al.[59] the BP training algorithm has been the most significant development in the growth of artificial neural networks and the BP networks have become the most well-known networks in technological applications .

3.2.1 The BP network architecture

The typical BP network architecture shown in Figure 3.1 is an example of a feedforward network with one or more hidden layers of processing units between the input and output layers of processing units. The duty of the first layer of processing units is to distribute the components of an input signal into the network and the characteristics of these units are different from the other processing units in the network. The rest of the processing units in the network perform the same functions as the processing unit in the elementary perceptron model mentioned in the previous chapter except that the sigmoid function is employed instead of a binary or bipolar function. When BP networks are used for function mapping, the activation function of the processing units in the output or the last layer can be changed to be the linear activation function from which the output is equal to the sum of the inputs. Usually the BP networks have the feedforward structure, however, the so-called 'recurrent' BP networks can also be found in the literature[62,70]. When building a BP network, one must decide on the number of layers and the number of processing units in each layer. For the first or input layer and the last layer or output layer, the number of processing units is prescribed by the dimensions of the input vectors or target vectors. However, general guidance concerning how many hidden layers and how many processing units in each hidden layer are needed, are still not available. In general, this depends on the complexity of the problem to be solved by the network, the required accuracy of the solution, the activation function used in the processing units and the characteristics of the training data. Although there are some rough guides for the two problems mentioned in the references [46,62,72], the trial-and-error method is still extensively employed.

3.2.2 The BP network learning algorithm

The fundamental idea of the BP learning algorithm is based on the steepest-descent searching technique often used in solving the problems of unconstrained optimisation. A general unconstrained minimisation problem can be considered as: to find an vector $\vec{x}$ that minimises an objective function $E(\vec{x})$. Then the steepest-descent technique transforms the optimisation problem to a set of dynamic equations which in the discrete-time version can take the form

$$\vec{x}(t+1) = \vec{x}(t) - \alpha(t)\nabla E(\vec{x}(t)) \qquad (3.1)$$

where $\alpha(t)$ is a positive parameter called the learning rate and $\nabla E(\vec{x}(t))$ is the gradient of the objective function. The learning rate is usually determined by minimising the function $\psi(\alpha) = E(\vec{x}(t) - \alpha\nabla E(\vec{x}(t)))$.

When applying the steepest-descent searching technique in the multi-layer perceptron network for adjusting weights, we need to define the objective function, which is often called the error function in neural network literature and can be expressed as

$$E^p = \frac{1}{2}\sum_{j=1}^{n}(t_j^p - o_j^p)^2 \tag{3.2}$$

where p stands for the pth input vector and j represents the jth output processing unit.

t_j^p is the target value of the jth output processing unit for the pth input vector.

o_j^p gives the actual output value of the jth output processing unit for the pth input vector.

Now, if we regard the connecting weights as a vector $\vec{w}$ and substitute it for the vector $\vec{x}$ in the above equations, a set of equations can be obtained in the form of the so-called least square technique. It is important to note that, before the BP algorithm was proposed, the adaptive equation (3.1) could only be employed for adjusting the weights connected to the processing units in the output layer and there was no way to apply this equation to adjust the weights located between the hidden layers. This meant that neural network could have only two layers, the input and the output layers, and no hidden layer was allowed. This limitation in the neural network structure was a serious defect which hindered the development of the neural network applications for a long time. Therefore, the importance of the BP learning algorithm becomes clear, since it generalised the least-square technique making it possible to adjust the hidden weights and to add hidden layers into the elementary perceptron network. Details of the derivation of the BP algorithm can be found in reference [59].

To train a BP network, sets of training data $(\vec{x}, \vec{t})$, which contain an input or exemplar vector $\vec{x}$ and a desired or target vector $\vec{t}$, must be prepared and then presented to the neural network one by one. The training session can be divided into three phases, which are signal feedforward, error feedback and weight-updating. The first phase starts after a set of training vectors have been presented to the BP network. The stimulus of the input vector $\vec{x}$ will be propagated through the network and will generate an output vector at the last layer. This output vector is compared with the preset target vector $\vec{t}$ to produce an error vector. In the second phase of the training session, the error vector in the output layer is fed backward into the network for the calculation of the local error vector in each individual layer. The second phase is complete when the error vector reaches the first layer. In the last training phase each component of the weight vector is updated according to an adjusting quantity which is a function of the corresponding local errors. Up to now, the first set of training data has finished its training session. Succeeding sets of training data are subsequently presented to the network following the same training procedure. The training session can be terminated when the value of the error function reaches an acceptable limit. The steps for the training session can be summarised as follows[46,59]:

(1) Set up a network that is suitable for a specified application.

(2) Initialise weights and thresholds to small random values.

(3) Choose an activation function, for example f(x)=1/(1+exp(-x)).

(4) Present training data sets $(\vec{x}, \vec{t})$ to the network. $\vec{x}$ is the input vector and $\vec{t}$ is the target vector. Calculate the output of each processing unit and the output error of each output processing unit.

(5) Update each component of the weight vector according to the following equations.

$$w_{mn}(t+1) = w_{mn}(t) + \Delta w_{mn}(t) \tag{3.3}$$

$$\Delta w_{mn}(t) = \alpha(t)\delta_n(t)o_m(t) \tag{3.4}$$

where $w_{mn}(t)$ is the weight between the processing unit m and the processing unit n at time t. $\alpha(t)$ is the learning rate at time t. $\delta_n(t)$ represents the error term of the processing unit n at time t. $o_m(t)$ stands for the output of processing unit m at time t.

The error term $\delta_n(t)$ has different interpretations for the processing units in the output layer and the processing unit in the hidden layers. If the unit is an output unit in the output layer, then

$$\delta_n(t) = f'(\sum_m w_{mn}(t)o_m(t)) \cdot (d_n(t) - o_n(t)) \qquad (3.5)$$

where $f'(*)$ is the derivative of the activation function with respect to *.
$d_n(t)$ is the target value of the processing unit n at time t.

If the processing unit n belongs to a hidden layer, then

$$\delta_n(t) = f'(\sum_m w_{mn}(t)o_m(t)) \cdot \sum_u \delta_u(t) w_{nu}(t) \qquad (3.6)$$

where u represents all processing units in the layer above the processing unit n except the threshold(bias) unit.

3.2.3 Some issues related to BP learning

There are still several important issues that need to be discussed. Generally speaking, the learning rate $\alpha(t)$ at equation (3.4) is a function of time and in the real steepest-descent searching technique it is found by minimising the function $\psi(\alpha) = E(\vec{w}(t) - \alpha \nabla E(\vec{w}(t)))$. However, in the neural network application, its value is guessed by the user and must be restricted to a positive real number less than 1. Large values of learning rate may speed up the learning speed but could cause an unstable learning process which may not converge to the required solution. Although small values of learning rate can achieve the goal of smoothly minimising the error function, the learning speed can be very slow. There are some techniques that can be applied to improve the learning speed of the BP algorithm[74-76]. The technique used in this thesis is acquired from experience of training the BP networks and reference[55]. The idea of this technique is rather simple. When the learning errors become unstable, or when the ratio of learning error of the previous learning cycle to the current learning cycle is greater than a pre-set value, or when the number of the training cycles is getting very large, the learning rate is reduced. Unfortunately, the exact time to change the learning rate is still not certain. In some cases, the improvement in convergence speed brought about by reducing the learning rate was very limited or even worse and it was necessary to change the learning rate again. However, the performance of this changing learning rate scheme was considered to be satisfactory.

In theory, we can apply the BP algorithm to train a multi-layer perceptron network to perform function mapping with desired accuracy. In practice, the algorithm itself will not guarantee that the final solution of the trained weight vector reaches the global minimum point of the error function. Frequently, during the training process, the weight vector can be trapped into a local minimum point and the final results do not reach the desired accuracy. To improve this problem, a momentum term is added to the weight updating equation (3.4) and the new updating quantity Δw_{mn} will take the form as

$$\Delta w_{mn}(t) = \alpha(t)\delta_n(t)o_m(t) + \beta(t)\Delta w_{mn}(t-1) \qquad (3.7)$$

$$\text{where } \Delta w_{mn}(t-1) = w_{mn}(t) - w_{mn}(t-1)$$

The weight updating scheme used in the training session mentioned above is called pattern learning. The name comes from the adjustment of the weight vector which is made once a set of training data completes its training session. An alternative weight adjusting scheme is called batch learning[62]. In this learning scheme the Δw_{mn}

for each input vector is accumulated and the weight updating is carried out only after the whole training data sets have finished the training session. Therefore, the total adjusting quantity of each component of the weight vector is the sum of the adjusting quantity Δw_{mn} of each weight component associated with each individual training data. The pattern learning scheme has been adopted as the standard technique in this thesis.

3.2.4 Examples of the application of BP networks

Two examples are given in this section to demonstrate different ways in which the BP networks can be used. The first example, a pattern classification problem, shows that the BP network is capable of handling the three dimensional exclusive-or problem which is linearly unseparable. The second example is an identification problem in which a feedback BP network is shown to be able to accurately simulate a nonlinear system.

The eight sets of training data with their target outputs for the exclusive-or problem are shown in the Table 3.1. The training output of each set of input data is also shown in the last column of the table. A three-layer network which has three input units in the first layer, three hidden processing units in the hidden layer and one output unit in the last layer is used for this demonstration. After training, the network can correctly separate the input data into two different classes. The mean squared errors (mse), which is the sum of errors obtained using equation 3.2 divided by the pattern numbers, with respect to the number of training cycles using BP networks with different numbers of hidden processing units are shown in Figure 3.2. For the purpose of comparison, the results from other networks with different hidden units in their hidden layer are also included. The legends h3, h5 and h10 shown in the figure are the number of units in the hidden layer. The results show that the training errors for each network improve rapidly in the early stage of the training session but later the rate of convergence becomes very slow. Also from the figure, it can be seen that the number of hidden processing units can influence the convergence rate as well as the error limits. The trained three-layer neural network, tested with untrained data sets, together with test results are shown in the Table 3.2.

The second example demonstrates that the BP network can be employed as an identification model for a non-linear system. According to the reference[77], there are two different approaches for identifying a system, namely parallel model and series-parallel model. Figure 3.3 shows these two identification approaches. The major difference between these approaches is that in the series-parallel approach the output from the identified system is fed back to the neural network model. In the parallel approach, the output from the neural network model is fed back to the model itself. The system to be identified in this example is governed by the difference equation (3.8)

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u) \quad (3.8)$$

where the input $u(k) = \sin(k2\pi / 250)$.

The BP network used in this example is shown in Figure 3.4. The structure of this network, which is different from the one used in the first example, has two hidden layers instead of one. In addition, the second network has two feedback inputs from the system output and the processing unit in the output layer is a linear unit from which the output is simply the sum of the outputs of the processing units in the previous layer. The final difference is the activation function used in this network which is

$$f(x)=1/(1+\exp(-x))-0.5 \quad (3.9)$$

In the first example the activation function was $f(x)=1/(1+\exp(-x))$. The activation function $f(x)=1/(1+\exp(-x))$ can only output values between 0 and 1, but the outputs for a system could be any real number. To identify the system, the neural network model must be able to generate an output of any real number. Therefore, an activation function capable of producing positive and negative values is needed and naturally the processing units located in the last layer have to be linear. This activation function has been used here, although the hyperbolic tangent function is the popular activation function used extensively in the literature. Later, in chapter 5, the hyperbolic tangent function is also examined(3.9).

The training results for this example are shown in Figure 3.5 for the series-parallel model and in Figure 3.6 for the parallel model. Owing to the scale of the figure, the differences between the system output and the neural network output are hidden, the training errors are plotted in Figure 3.7 and Figure 3.8 respectively for reference. The errors of the series-parallel identification approach are smaller than those generated by the parallel approach. The reason for this is quite clear, since in the parallel identification approach the outputs of the neural network model are fed back to the model itself and the errors are also fed back to the network. In the series-parallel approach, the input signals to the network do not include errors. In spite of the slightly bigger errors, the result of the parallel identification approach shown in the Figure 3.6 is still comparable with the result of the series-parallel approach in the Figure 3.5. Although the series-parallel approach seems to be more attractive for the identification application, the parallel identification approach is still very important in condition monitoring applications. Here neural networks are used as the reference model for the healthy system and the fed back signals should be from the neural network itself and can not be from the monitored system in case the monitored system is faulty.

3.3 Kohonen's self-organizing map(SOM)[55,78]

The multi-layer perceptron neural network discussed in the last section is categorised in a group of artificial neural networks which use a supervised learning scheme; an external learning target must be presented to the network to teach the network how to adjust the weights. However, examining our learning processes, it is quite apparent that in some cases we do not need a teacher to teach us how to learn and we can learn by ourselves. This unsupervised learning idea is adopted in the self-organizing map learning algorithm. The self-organizing map is also motivated by the biological evidence that our brain has maps corresponding to different parts of our body, therefore a particular part of neural cells respond strongly to some sort of external stimuli and not strongly to others. Similarly, the units in the self-organizing maps are specially tuned to respond to various input signal patterns and a map, which represents the internal information spatially, is formed automatically through an unsupervising encoding algorithm. Despite this general unsupervised learning characteristic, a supervised learning scheme, usually called learning vector quantization(LVQ), can also be applied to networks when using them for pattern recognition or other decision making processes. There are three different strategies of supervised learning, namely LVQ1, LVQ2 and LVQ3.

3.3.1 The SOM architecture

The architecture of a basic self-organizing map is composed of two layers of processing units as shown in Figure 3.9. Each unit in the input layer is connected to each unit in the upper two-dimensional layer and each connection has a weight which needs to be adjusted in the tuning process. The number of units in the input layer is equal to the dimensions of the input patterns. The number of units in the second layer, which generates the map output(also called a feature map), is determined by the topological dispersion of the classes present in the input patterns. A larger number of units may be necessary if the clusters of the feature map are to be spatially separated by a larger distance. After the tuning process is finished, each class of input patterns forms a cluster on the second layer of the network. An extra layer, which may be called the class output layer, can be added to the basic map architecture in order to send out information about the classes of the input patterns when the map is used for the purpose of pattern classification. In this case, the weights w_{ij} connected from the two-dimensional layer to the class output layer are pre-set to be 1 or 0. The weight w_{ik} is 1, if the output of the unit i in the two-dimensional layer is known to respond to the pattern vectors belonging to a given class, and the unit k in the class output layer is also assigned to represent the same class, otherwise the weight is set to 0. This new architecture of the self-organizing map will result in the hard partitions of the classes and is shown in Figure 3.10.

3.3.2 The SOM learning algorithm

A competitive learning scheme for adaptively updating the weights is employed for tuning the network. Before starting the unsupervised learning process, all weights must be initialised to be small random real numbers. Then a pattern vector $\vec{x}(t)$, chosen from a queue of training or reference vectors, is presented to the network. For each second layer processing unit a measure d_i is calculated using:

$$d_i = \left\| \vec{x}(t) - \vec{w}_i(t) \right\| \tag{3.10}$$

where d_i is the measure of the ith unit in the map layer.

$\|*\|$ is an arbitrary norm of *.

$\vec{w}_i(t)$ is the weight vector of the ith unit in the second layer at time t.

The winning unit in the second layer is chosen by the following criterion

$$\left\| \vec{x}(t) - \vec{w}_w(t) \right\| = \min \left\| \vec{x}(t) - \vec{w}_i(t) \right\| \quad \forall\, i \qquad (3.11)$$

where $\vec{w}_w(t)$ is the weight vector of the winning unit at time t.

$\min\|*\|$ is the minimum norm of *.

In order to encode the feature of every individual reference vector evenly in the output map, the process of updating the weights is carried out according to equation (3.12) and equation (3.13). In the equations, the updating neighbourhood is the user defined area surrounding the winning units and used for adjusting the weights.

$$\vec{w}_i(t+1) = \vec{w}_i(t) + \alpha(t)(\vec{x}(t) - \vec{w}_i(t)) \quad \forall\, i \in N_c(t) \qquad (3.12)$$

$$\vec{w}_i(t+1) = \vec{w}_i(t) \qquad \forall\, i \notin N_c(t) \qquad (3.13)$$

where $0<\alpha(t)<1$ is the tuning rate decreasing monotonically during tuning course.

$N_C(t)$ is a shrinking neighbourhood surrounding the winning unit, see Figure 3.13.

Although the norm used in the above equation is rather general, it often takes the form $\sum_{j=1}^{m}(x_j(t) - w_{ji}(t))^2$, where x_j is jth component of the input pattern vector and w_{ji} is the jth component of the ith weight vector. The physical meaning of the weight updating equation (3.12) and (3.13) is easily understood when compared with Figure 3.12. The weight vector of the winning unit and its neighbouring units are moved in the direction of $\vec{x}(t)$-$\vec{w}_i(t)$ gradually with a small amount of the vector, $\alpha(t)[\vec{x}(t)$-$\vec{w}_i(t)]$ and after each learning cycle the weight vectors of the winning units are moving closer to the input pattern vectors. Eventually, the weight vectors of winning units will converge to the pattern vectors. An alternative to the equation (3.10) is the inner-product measure of the similarity between the input pattern vector and each individual weight vector and can be expressed as $d_i = \vec{x}(t) \bullet \vec{w}_i(t)$. In this case the winning unit will be the one with the maximum value of d_i rather than the minimum value as in the equation (3.11). The weight updating equations are

$$\vec{w}_i(t+1) = \frac{\vec{w}_i(t) + \alpha_1(t)\vec{x}(t)}{\left\| \vec{w}_i(t) + \alpha_1(t)\vec{x}(t) \right\|} \quad \text{if } i \in N_c(t) \qquad (3.14)$$

$$\vec{w}_i(t+1) = \vec{w}_i(t) \quad \text{if } i \notin N_c \qquad (3.15)$$

where $0<\alpha_1(t)<\infty$ is the learning rate.

If the network is used for pattern recognition then fine tuning, which is a supervised learning process for adjusting connecting weights, must be enforced after using the self-organizing algorithm for coarse learning. Otherwise, a confusion of classification between different classes along class boundaries is likely to happen. There are three different algorithms[55] for the fine tuning process. LVQ1 used in this thesis, and the fine tuning process can be achieved by using equation (3.16) to equation (3.18).

$$\vec{w}_i(t+1) = \vec{w}_i(t) + \alpha(t)[\vec{x}(t) - \vec{w}_i(t)] \quad \text{if correct classification} \qquad (3.16)$$

$$\vec{w}_i(t+1) = \vec{w}_i(t) - \alpha(t)[\vec{x}(t) - \vec{w}_i(t)] \quad \text{if incorrect classification} \qquad (3.17)$$

$$\vec{w}_i(t+1) = \vec{w}_i(t) \quad \text{otherwise} \qquad (3.18)$$

Briefly, this is a reward-punish algorithm. If a winning unit for an input pattern vector is the correct class then the weight vector of this unit is rewarded by strengthening the weight vector using equation (3.16). On the other hand, if the winning unit and the input pattern vector do not belong to the same class the weight vector of the winning unit will be punished by subtracting an amount from it according the equation (3.17). If the winning unit does not belong to any existing class, then the weight vector of the winning unit will be kept intact. Hopefully, after the fine tuning process has been completed the boundaries of different pattern classes can be distinctly separated and the chance of misclassification will be minimised.

3.3.3 Some issues regarding the learning of the SOM

The learning in the self-organizing maps is a stochastic process, in which a large number of iterations is necessary in order to obtain accurate results. Kohonen[55] suggested that for good statistical accuracy, the number of iterations must be at least 500 times the number of network units. This criterion means that the learning time for a network with a large number of second layer processing units could be very long. However, Kohonen mentioned that sometimes "fast learning", which requires far less iterations, are enough for a map to complete the learning session. The learning rate α (t) is an important parameter pertaining to the success of the learning. The ordering of the map occurs in the initial period and the learning rate should be large. The remaining iterations are only needed for the fine adjustment of the map, therefore the value of α (t) should be small. Kohonen suggested some guides for the values of $\alpha(t)$. For approximately the first 1000 iterations, the value of $\alpha(t)$ must close to 1, thereafter decreasing monotonically. The type of the monotonically decreasing function for $\alpha(t)$ is immaterial and it can be linear, exponential or inversely proportional to time. The final issue which is extremely important to the learning results, is the updating neighbourhood Nc(t). The shape of the neighbourhood can take different patterns, for example rectangular, hexagonal or circular areas are common. The size of the updating neighbourhood Nc(t) starts with a very wide area and gradually shrinks to cover only the winning unit during the period of learning. If the initial neighbourhood is too small, the final map will not be ordered globally. To avoid this happening, the initial area of this neighbourhood should be very large, and can be larger than the size of the network itself.

There are some minor imperfections about the learning algorithm of self-organizing maps which should be mentioned.

(a) The final results of the self-organizing map are affected by the sequence of the input pattern sets and the strategies for the learning rate $\alpha(t)$.[79] If we change the sequence of input data or use different strategies for $\alpha(t)$, often the outcomes are different.

(b) The algorithm of self-organizing maps do give a map of clusters of the exemplar patterns if the learning follows the general rules of learning mentioned in the previous paragraph. But we do not know what sort or type the map is or what sort or type of points the learned weights are. Regarding this point, Pal et al[79] criticised the LVQ model because it is not driven by a well specified clustering goal.

(c) The learning session only stops after the learning rate runs out or is forced to terminate by the simulation program. The learning algorithm does not set up any criterion for the learning to stop and the LVQ often passes the optimal clustering solution[79].

3.3.4 Examples of applications of the SOM

Two examples are given in this section. In the first example sixteen sets of (x,y) values are use to test the self-organizing map network to see if it can separate them into corresponding groups. Also, by comparing the initial weights with the final learned weights of the network we can understand that the network indeed encoded the input pattern vectors into its weights. The second example is taken from the reference [55], originally for testing the author's program, to show that the map can perform mapping of a taxonomic graph of numerical data.

The data sets for the first example are listed in the Table 3.3 and the learned maps are shown in Figure 3.13. There are two sets of results in the figure, the set (a1)(a2) and the set (b1)(b2). They are produced by the same learning strategies but with different sets of initial weight vectors. In each figure of the Figures 3.13(a1) and 3.13(b1), the winning units for the data sets listed in Table 3.3 are labelled with the data set numbers. Also, for comparison, in each of the figures of 3.13(a2) and 3.13(b2), the values of the data set corresponding to the data numbers in the figures 3.13(a1) and 3.13(b1) are shown. From these figures it can be seen that the data sets belonging to different classes are grouped together in the maps. Furthermore, if the locations of the winning units and the exemplar data sets, shown in Figure 3.13(b1) and (b2), are checked then it will be seen that these locations actually match the relative locations of the data sets in the x-y plane. For convenience, the relative locations of the exemplar data sets in the maps(x-y co-ordinates) are also presented in the figures 3.13(b1) and 3.13(b2). It is interesting to note that, because of the different initial weight vectors, the final locations of the exemplar data sets in the map 3.13(b2) can be obtained by flipping the map 3.13(b1) vertically.

In order to understand the learning results we have to go one more step to check the weight vectors inside the network. First we need to compare the initial weight vectors with the learned or final weight vectors. Both of the initial and final weight vectors are listed in Table 3.4(a) and (b) for different initial weight vectors. For example, the unit in the map with co-ordinates (0,0) in Figure 3.13(a2) has the initial weight vectors (0.138062,0.390564) in Table 3.4(a). After a learning session, the weight vector has been changed to (-2.000000,-2.999999), which represents the input pattern vector (-2,-3) labelled set number 4, in Figure 3.13(a1). This is not a coincidence. It is because the initial weight vector has been moved to the input pattern vector step by step using the learning rule (equation (3.12) and Figure 3.12). Therefore, the output from unit (0,0) in the map shows that the input pattern vector is a member belonging to class 1. By the same token, we can check each of the learnt vectors to see which unit in the output map will respond to which input pattern vectors. The same explanations are also applicable to Figures 3.13(b1) and 3.13(b2).

In the second example, sets of artificial data called pattern vectors, each having five components, are produced arbitrarily to represent the letters from 'a' to 'z' and the numbers from 1 to 6, as shown in Table 3.5. These vectors are then used for the learning of the self-organizing map. The learning results are shown in Figure 3.14(a) and (b). The two results are slightly different because the initial weight vectors are different. This demonstrates that the final learning results of the map are learning parameter related. The dotted lines in the two figures are drawn for the ease of comparing the locations of the encoded patterns with the minimal spanning tree, shown in Figure 3.15. This depicts the similarity relations of the data sets in Table 3.5. The absolute locations of the labels in the maps are different from the results in the reference paper [55] because of the different initial weight vectors and Nc(t), although the relative locations of the labels are similar.

3.4 Adaptive resonance theories

A human brain can simultaneously accept new information while still retaining a tremendous amount of previously learned knowledge. In short, we can learn new things without losing old memories. However, for most artificial neural networks, an attempt to add new information by adding new training or exemplar vectors to an already trained neural network frequently destroys its existing long-term memory structure and consequently loses everything encoded in the network. This dilemma is referred to as the stability-plasticity problem. Neural networks constructed by applying adaptive resonance theories, often called ART networks, are ingeniously devised so that the stability-plasticity problem is solved and the learning course can be completed in a very short period of time. In the algorithms used in ART networks for categorising sequence pattern inputs, unlike the self-organising map, the sequence is not important to the final results. This makes their performance closer to that of the human brain than other networks. The ART family has three members. ART1[50] is designed for binary input patterns, ART2[51] is capable of handling binary as well as analogue input pattern vectors, and ART3[52] is the newest version, into which the concepts of transmitter dynamics which occur in biological neurons is infused.

3.4.1 ART architectures

All of the ART family are based on the same fundamental architecture as shown in Figure 3.16. The input processor, shown by the dotted box, and the arrowhead line from the dotted box to the orienting subsystem are special for the ART2 network used in this thesis, therefore, it can be neglected for the time being. In addition, for clarity, all the gain control elements are omitted in this figure. Two subsystems which are called the attentional subsystem and the orienting subsystem are shown. The bottom-up and top-down LTM(long term memory) traces, or adaptive weights, are elements equivalent to the weights in other types of artificial neural networks. The attentional subsystem is in charge of processing the inputs, controlling the signal flow and generating the outputs. The orienting subsystem is responsible for checking mismatches between the input patterns and the output patterns and resetting the output units if mismatches occur. There are two layers of processing units in the attentional subsystem, the lower layer is called the comparison layer or feature presentation field and the higher layer is named the recognition layer or category representation field. Each unit in the recognition layer symbolises a group of input pattern vectors and a single group or several groups may stand for a class of pattern vectors presented to the ART network. In addition, units in the recognition layer can be added and the only restriction is the available memory size of the computational system, if the ART networks are used for simulation.

3.4.2 The process of categorisation in ART networks

For the majority of artificial neural networks, a learning or training phase is essential for these networks to encode necessary information into their weight vectors before they can be used. However, the special designed architecture of the ART network makes it possible to perform the encoding and decoding tasks simultaneously, namely, the ART networks can learn new patterns when they first see them. The following description of the process of categorisation in ART networks does not cover all details and only those main steps will be mentioned.

Referring to Figure 3.17(a), when a pattern **i** is presented to the ART network, responding to the stimulus caused by the input pattern vector, the feature presentation field F1 sends out a signal pattern **r** to the category representation field F2. Before the signal pattern **r** reaches the F2 field , the signal **r** is gated by the bottom-up LTM and is transformed to another signal **s**. Each F2 processing unit, whose output represents a pattern class, if it has been encoded in the past, acts as an instar unit(Chapter 2) and a competition among all F2 units is carried out after the arrival of the pattern **s**. This competition allows only one winning unit to produce its output and the rest of units are suppressed, this is called the winner-takes-all scheme. Subsequently, an output pattern **t**, compressed and stored in the winning unit, will be retrieved and sent back to the F1 field. In Figure 3.17(b), the winning unit is shown by a filled circle and the compressed pattern is portrayed in the boxes just above the F2 field. Now, the winning unit in the F2 field behaves as an outstar unit. The pattern **t** is gated as well by the top-down LTM and transformed to the pattern **u**. The whole course undertaken up to this point can be called the searching stage, shown in the figure (a) and (b). The following course is called the matching stage, shown in the figure (c).

To start the matching stage, the top-down pattern **u** and the input pattern **i** are sent to the orienting subsystem and a similarity test between the two patterns is carried out. If these two patterns are similar enough, evaluated by a pre-set criterion which is symbolised by the circle with 'vig' inside, then the LTMs associated with the winning unit will be updated using the instar and outstar learning rules. On the other hand, if mismatch does happen, then the F2 field will be reset and the present winning unit will be disabled and thereafter kept from joining the competition as long as the same input pattern is the current input. The disabled unit is shown in Figures 3.17(c) and (d) marked with a cross. The current input pattern is once more presented to the F1 field to restart the searching and match-testing processes, shown in the figure (d). These processes will proceed until a best match pattern is found or all encoded units in the F2 field are exhausted. In the latter case, an uncommitted F2 unit is assigned to represent the new group of patterns. As soon as the current input pattern has found the best-match pattern in the F2 field, the searching-matching cycles repeat again for a new input pattern.

3.4.3 Details of the searching-matching process in an ART2 network[51]

In the previous section the general searching-matching processes in ART networks were described. In this section, the details of these processes in the ART2 network will be illustrated. Although there are alternative of ART2 architectures, only one is referred to in this thesis. This is shown in Figure 3.19 and the connections between the F1 field and the F2 field are shown in Figure 3.18. In Figure 3.19 the bold letters represent not only the processing unit at that special point but also the vectors at the same point. From the pattern vector input point to the point **i**, the input vector is normalised and fed back to the input point, thereafter the signal proceeds and resonates in the F1 field according to the activity equations shown below. For ease of understanding all equations take the component forms instead of vector forms.

$$w_i = i_i + au_i \tag{3.19}$$

$$x_i = \frac{w_i}{e + \|w\|} \tag{3.20}$$

$$v_i = f(x_i) + bf(q_i) \tag{3.21}$$

$$f(y) = \begin{cases} \dfrac{2\beta y^2}{y^2+\beta^2} & \text{if } 0 \le y \le \beta \\ y & \text{if } y \ge \beta \end{cases} \tag{3.22}$$

$$u_i = \frac{v_i}{e + \|v\|} \tag{3.23}$$

$$p_i = u_i + \sum_j g(t_j) n_{ji} \tag{3.24}$$

$$q_i = \frac{p_i}{e + \|p\|} \tag{3.25}$$

where $\|\mathbf{p}\|$ is the norm of the vector **p**.

p_i is the ith component of the vector **p**.

t_j is the short-term memory activity of the jth unit in the F2 field.

n_{ji} is the adaptive weights connected from the jth F2 unit to the ith F1 unit.

$g(t_j)$ is the output of the jth unit in the F2 field.

a,b,e and β are parameters.

After the resonance has stabilised in the F1 field, the output vector **p** is gated by the bottom-up adaptive weights and summed before entering the F2 field, shown in Figure 3.18(a). Following this the competition within the F2 field generates only one winning unit labelled J. This winning unit is chosen by the equation (3.26) and an output $g(t_J)$=

δ is produced by it. As a result the general equation (3.24) turns out to be the equation (3.27), shown in Figure 3.18.

$$\Gamma_J = \max\left\langle \Gamma_k = \sum_i p_i m_{ik} \right\rangle \quad \forall\, k \in F2 \qquad (3.26)$$

$$p_i = \begin{cases} u_i + \delta\, n_{Ji} & \text{if the Jth F2 unit is active} \\ u_i & \text{if F2 is inactive} \end{cases} \qquad (3.27)$$

where m_{ik} is the adaptive weights connected from the ith F1 unit to the kth F2 unit.

J is the label of the winning unit in the F2 field.

The output pattern retrieved from the winning unit is now subjected to a match test to check if the input pattern really matches it. For this purpose a new vector **r** is formed in the orienting subsystem by equation (3.28). Then the orienting subsystem uses equation (3.29) as a criterion to evaluate the result of the searching process.

$$r_i = \frac{i_i + cp_i}{e + \|\mathbf{i}\| + \|c\mathbf{p}\|} \qquad (3.28)$$

$$\sigma = \frac{\rho}{e + \|\mathbf{r}\|} \qquad (3.29)$$

where c and ρ are parameters.

The parameter ρ is called the vigilance parameter and its value measures the degree to which the system discriminates between different groups of input patterns. For a given set of patterns to be categorised, the larger the vigilance parameter the finer is the discrimination between groups. The value of σ indicates the result of matching test. σ >1 means that the similarity between the winning pattern and the input pattern does not satisfy the pre-set criterion and the group represented by the winning unit is discarded. A new searching-matching session starts automatically in the attentional subsystem as illustrated in the last section. Eventually, the input pattern will be assigned to a pattern group symbolised by a winning unit in the F2 field and the adaptive weights, bottom-up as well as top-down, associated to this winning unit will be updated following equation (3.30) in the fast learning mode or equation (3.31) and (3.32) in the slow learning mode.

$$m_{iJ} = n_{Ji} = \frac{u_i}{1-\delta} \tag{3.30}$$

$$m_{iJ}(t+1) = m_{iJ}(t) + g(\sum_i p_i(t) m_{iJ}(t))(p_i(t) - m_{iJ}(t)) \tag{3.31}$$

$$n_{Ji}(t+1) = n_{Ji}(t) + g(\sum_i p_i(t) m_{iJ}(t))(p_i(t) - n_{Ji}(t)) \tag{3.32}$$

where $g(*)$ is the activation function for the units in the F2 field.

Finally, it is worthy of note that it is important to choose correct values for parameters following those constraints described in reference [51]. Should an incorrect value be chosen, the categorisation process could fail or become unstable.

3.4.4 Examples of pattern classification using the ART networks

Three examples relating to applications of ART networks in pattern classifications will be demonstrated. The first example concerns an ART1 network and the second and third examples concern the ART2 networks. The data set used in the first example is the same as the parity example shown in the section 3.2.3. The other two data sets are the same as those employed for the examples in the section 3.3.4.

The reason for using the same data sets is to provide some sort of comparison between the different kinds of networks applied extensively in solving the pattern classification problems.

The data sets for the first example and the results produced by using an ART1 network for categorisation are shown in the Table 3.6. The group numbers for the corresponding pattern vectors categorised by the ART1 network are shown in the last two columns of the table. If we carefully examine the input pattern vectors shown in the table, we will discover that some of the vectors are the subsets of some of the rest vectors. For example set number one is the subset of the set number 4,5 and 8. The ART1 has a characteristic that if the input vector is a subset vector of a previously encoded vector then the LTM of the encoded vector will be re-encoded by the subset vector and the original LTM will be washed away. This does no harm to the superset vector because it can re-build its LTM at the second cycle when it is presented to the network.

In the input pattern vector there is a zero vector which needs special treatment. Recalling the fundamental ART concepts mentioned in the last two sections, there is an important parameter called the vigilance parameter for evaluating the similarity of the input pattern vector and the winning pattern vector. In the ART1 networks we need to use the input vector as the denominator to generate a value in order to compare this with the vigilance parameter. If the input vector is a zero vector then it is impossible to produce this comparison value. To solve this problem, in the ART1 program the zero vector is arbitrarily assigned to the group zero. Now, if we check Tables 3.7(a) and 3.7(b), we can find out that after the zero vector is assigned to the category zero, the LTM of the vector number one is replaced by the zero vector. At the second presentation, the pattern set number one takes the place(category number 3) for its superset vector(set number 3) and its superset vector immediately settles down at the category number 6 which previously belonged to its superset vector(vector number 8). In turn the vector set number 8 finally rests at the category 7.

For this example the categorisation process is completed in just two presentation cycles by the fast learning scheme. Actually, the data sets were tested with more than two presentation cycles to see whether or not the network had settled down. The bottom-up and the top-down LTM for different categories are shown in Tables 3.7(a) and 3.7(b) respectively. Some uncommitted units in the representation field F2 are also shown in the tables. As mentioned before, we can add as many F2 units as we like to the F2 field to increase the capacity of the network for adopting new categories. The initial LTM, which are different for the bottom-up and top-down ones, are also shown in the cells of uncommitted categories. The patterns encoded in the network can be seen clearly from the top-down LTM. For instance, referring to the Table 3.6 the pattern vector number 5 has the components (1,0,1) and is encoded in the category number 4. To see if this vector (1,0,1) is indeed encoded in the category 4, we can check the bottom-up LTM in the Table 3.7(a) group number 4, which is (0.4,0.0,0.4) and the top-down LTM in the Table 3.7(b) group number 4, which is (1.0,0.0,1.0). Other pattern vectors listed in the Table 3.6 can also be examined in the same way.

The data sets given in Table 3.8 and Table 3.10 are used for two examples to demonstrate the capacity of ART2 networks. The pattern vectors shown in Table 3.8 are the same as those in Table 3.3 except that each number in Table 3.3 is expressed in two numbers in the Table 3.8. For example, the positive number 2 in the Table 3.3 is 0.0 and 2.0 in the Table 3.8 and the negative number -2 is 1.0 and 2.0, namely a sign representative number is added, because of the need of positive inputs in the ART2 networks.

The final results for the first ART2 example are shown in the last two columns of the Table 3.8 and the corresponding LTM are listed Table 3.9(a) and (b). The uncommitted units in the F2 field have the same initial bottom-up LTM and the same initial top-down LTM respectively, however, for those committed units the LTM are the same for the same unit no matter bottom-up or top-down. The influence of the vigilance parameter in categorisation can be understood by comparing the categories corresponding to different pattern vectors in the last two columns in the Table 3.8. The data sets and results of the second example using the ART2 network are shown in Table 3.10. The results of the LTM are omitted because they are similar to the previous example.

3.5 Closure

In this chapter three different types of artificial neural networks, which will be used in this thesis for hydraulic power system condition monitoring, were discussed. Examples were employed to show the capabilities of these artificial neural networks and the computer programs needed for simulating these examples were coded in 'C' language and executed on a 486-DX33 personal computer. These examples and the original theories show that back-propagation multi-layer(BP) networks are good for both system modelling and pattern recognition and self-organising maps(SOMs) and that ART networks are suitable for unsupervised pattern recognition. In addition, ART networks are known not to suffer from the so-called elasticity-plasticity problem which happens will most of artificial neural networks. In condition monitoring systems, precise reference models are often required and using BP networks to build reference models can be one of the best choices for system modelling. Traditional fault classification normally rely on human decision making which are usually fulfilled by using expert systems.[13,27,30] In this study, fault classification will be carried out using pattern recognition techniques derived from artificial neural networks.

Table 3.1 Training data and outputs for the parity example.

no.	training data sets			target	output
1	0.0	0.0	0.0	0.9	0.9
2	0.0	0.0	1.0	0.1	0.1
3	0.0	1.0	0.0	0.1	0.1
4	1.0	0.0	0.0	0.1	0.1
5	0.0	1.0	1.0	0.9	0.9
6	1.0	0.0	1.0	0.9	0.9
7	1.0	1.0	0.0	0.9	0.9
8	1.0	1.0	1.0	0.1	0.1

Table 3.2 test data and test results for the parity example.

set no.	test data sets			output
1	0.0	0.1	0.1	0.78
2	0.0	0.0	0.0	0.90
3	0.0	0.0	0.8	0.11
4	0.9	1.0	0.9	0.18
5	1.0	1.0	1.0	0.10
6	0.2	0.0	0.0	0.78
7	0.9	0.1	0.9	0.90
8	0.0	0.2	0.0	0.78
9	1.0	0.8	0.9	0.25
10	1.0	0.8	0.7	0.46

Table 3.3 Exemplar data sets for the first SOM example.

data no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x values	1	1	-2	-2	2	2	-1	-1	2	2	-1	-1	1	1	-2	-2
y values	3	-2	3	-3	3	-2	2	-2	2	-3	3	-3	2	-3	2	-2
classes	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

Table 3.4(a) The initial weight vectors and the learnt weight vectors associated with Figures 3.13(a1) and 3.13(a2).

(row,column)	initial weight vectors		learnt weight vectors	
(0 , 0)	0.138062	0.390564	-2.000000	-2.999999
(0 , 1)	0.025848	0.972809	-1.000000	-2.999999
(0 , 2)	0.518280	0.555511	0.085879	-3.000000
(0 , 3)	0.240845	0.294128	1.000000	-2.999999
(0 , 4)	0.848053	0.021545	2.000000	-2.999999
(1 , 0)	0.077454	0.435364	-2.000000	-2.000001
(1 , 1)	0.761261	0.156769	-1.000000	-2.000001
(1 , 2)	0.936890	0.927643	-0.085879	-2.000001
(1 , 3)	0.446411	0.460449	1.000000	-2.000001
(1 , 4)	0.666412	0.707581	2.000000	-2.000001
(2 , 0)	0.257751	0.564911	-2.000000	-0.171758
(2 , 1)	0.076996	0.275665	-1.000000	-0.171758
(2 , 2)	0.639221	0.050140	0.709539	0.477812
(2 , 3)	0.773407	0.666809	1.000000	0.171758
(2 , 4)	0.360748	0.070587	2.000000	0.171758
(3 , 0)	0.296692	0.233643	-2.000000	2.000001
(3 , 1)	0.067108	0.794708	-1.000000	2.000001
(3 , 2)	0.579407	0.242157	0.085879	2.000001
(3 , 3)	0.139160	0.656860	1.000000	2.000001
(3 , 4)	0.712952	0.279968	2.000000	2.000001
(4 , 0)	0.867798	0.421814	-2.000000	2.999999
(4 , 1)	0.027985	0.476959	-1.000000	2.999999
(4 , 2)	0.940155	0.061646	-0.085879	2.999999
(4 , 3)	0.486664	0.538574	1.000000	2.999999
(4 , 4)	0.238556	0.445251	2.000000	2.999999

Table 3.4(b) The initial weight vectors and the learnt weight vectors associated with the map in Figures 3.13(b1) and 3.13(b2).

(row,column)	initial weight vectors		learnt weight vectors	
(0 , 0)	-0.361938	-0.109436	-1.999998	2.999997
(0 , 1)	-0.474152	0.472809	-1.000002	2.999997
(0 , 2)	0.018280	0.055511	-0.056858	2.999999
(0 , 3)	-0.259155	-0.205872	1.000002	2.999997
(0 , 4)	0.348053	-0.478455	1.999998	2.999997
(1 , 0)	-0.422546	-0.064636	-1.999998	2.000003
(1 , 1)	0.261261	-0.343231	-1.000002	2.000003
(1 , 2)	0.436890	0.427643	0.056858	2.000001
(1 , 3)	-0.053589	-0.039551	1.000002	2.000003
(1 , 4)	0.166412	0.207581	1.999998	2.000003
(2 , 0)	-0.242249	0.064911	-2.000000	-0.113715
(2 , 1)	-0.423004	-0.224335	-1.000000	-0.113715
(2 , 2)	0.139221	-0.449860	0.883360	-0.437522
(2 , 3)	0.273407	0.166809	1.000000	0.113714
(2 , 4)	-0.139252	-0.429413	2.000000	0.113715
(3 , 0)	-0.203308	-0.266357	-1.999998	-2.000003
(3 , 1)	-0.432892	0.294708	-1.000002	-2.000003
(3 , 2)	0.079407	-0.257843	-0.056858	-2.000001
(3 , 3)	-0.360840	0.156860	1.000002	-2.000003
(3 , 4)	0.212952	-0.220032	1.999998	-2.000003
(4 , 0)	0.367798	-0.078186	-1.999998	-2.999997
(4 , 1)	-0.472015	-0.023041	-1.000002	-2.999997
(4 , 2)	0.440155	-0.438354	0.056858	-2.999999
(4 , 3)	-0.013336	0.038574	1.000002	-2.999997
(4 , 4)	-0.261444	-0.054749	1.999998	-2.999997

Table 3.5 Data sets for the second SOM example.

attributes(input pattern vectors)					labels
1	0	0	0	0	a
2	0	0	0	0	b
3	0	0	0	0	c
4	0	0	0	0	d
5	0	0	0	0	e
3	1	0	0	0	f
3	2	0	0	0	g
3	3	0	0	0	h
3	4	0	0	0	i
3	5	0	0	0	j
3	3	1	0	0	k
3	3	2	0	0	l
3	3	3	0	0	m
3	3	4	0	0	n
3	3	5	0	0	o
3	3	6	0	0	p
3	3	7	0	0	q
3	3	8	0	0	r
3	3	3	1	0	s
3	3	3	2	0	t
3	3	3	3	0	u
3	3	3	4	0	v
3	3	6	1	0	w
3	3	6	2	0	x
3	3	6	3	0	y
3	3	6	4	0	z
3	3	6	2	1	1
3	3	6	2	2	2
3	3	6	2	3	3
3	3	6	2	4	4
3	3	6	2	5	5
3	3	6	2	6	6

Table 3.6 Data sets and the results of categorisation using ART1 network.

data set number	input pattern vectors			group no. (1st cycle)	group no. (2nd cycle)
1	1	0	0	0	3
2	0	1	0	1	1
3	0	0	1	2	2
4	1	1	0	3	6
5	1	0	1	4	4
6	0	1	1	5	5
7	0	0	0	0	0
8	1	1	1	6	7

Table 3.7(a) The bottom-up LTM of the ART1 network for the first example.

gp. no.♠	bottom-up LTM of the first cycle			bottom-up LTM of the second cycle		
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.666667	0.000000	0.000000	0.666667	0.000000
2	0.000000	0.000000	0.666667	0.000000	0.000000	0.666667
3	0.400000	0.400000	0.000000	0.666667	0.000000	0.000000
4	0.400000	0.000000	0.400000	0.400000	0.000000	0.400000
5	0.000000	0.400000	0.400000	0.000000	0.400000	0.400000
6	0.285714	0.285714	0.285714	0.400000	0.400000	0.000000
7**	0.250000	0.250000	0.250000	0.285714	0.285714	0.285714
8*	0.250000	0.250000	0.250000	0.250000	0.250000	0.250000
9*	0.250000	0.250000	0.250000	0.250000	0.250000	0.250000

♠ : group number

** : uncommitted in the first cycle, committed in the second cycle

* : uncommitted

Table 3.7(b) The top-down LTM of the ART1 network for the first example.

gp. no.♠	top-down LTM of the first cycle			top-down LTM of the second cycle		
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000
2	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000
3	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000
4	1.000000	0.000000	1.000000	1.000000	0.000000	1.000000
5	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000
6	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
7**	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
8*	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
9*	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Table 3.8 The data sets with the category outputs of the second example.

input pattern vectors				labelled category	category (ρ=0.98)	category (ρ=0.99)
0.0	1.0	0.0	3.0	1	0	0
0.0	1.0	1.0	2.0	2	1	1
1.0	2.0	0.0	3.0	3	2	2
1.0	2.0	1.0	3.0	4	4	6
0.0	2.0	0.0	3.0	1	3	5
0.0	2.0	1.0	2.0	2	1	3
1.0	1.0	0.0	2.0	3	2	2
1.0	1.0	1.0	2.0	4	4	4
0.0	2.0	0.0	2.0	1	3	5
0.0	2.0	1.0	3.0	2	1	1
1.0	1.0	0.0	3.0	3	2	2
1.0	1.0	1.0	3.0	4	1	4
0.0	1.0	0.0	2.0	1	0	0
0.0	1.0	1.0	3.0	2	1	1
1.0	2.0	0.0	2.0	3	2	2
1.0	2.0	1.0	2.0	4	4	6

Table 3.9(a) The bottom-up LTM generated by ART2 networks using different vigilance values for the input pattern vectors listed in the Table 3.8.

bottom-up LTM (ρ=0.98)				bottom-up LTM (ρ=0.99)			
0.071697	3.522572	0.000000	9.358758	0.000000	3.458943	0.000000	9.382735
0.487253	4.334965	3.540942	8.272388	0.000000	4.123555	3.715506	8.318131
3.698423	4.497518	0.000000	8.129822	3.037617	5.084273	0.048271	8.057340
0.000000	6.879451	0.000000	7.257627	0.000000	6.666667	3.333333	6.666667
3.190798	5.892286	3.190798	6.702133	3.636764	3.636764	3.636764	7.766714
1.250000	1.250000	1.250000	1.250000	0.000000	6.879451	0.000000	7.257627
1.250000	1.250000	1.250000	1.250000	3.090680	6.181360	3.090680	6.533467
1.250000	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000
1.250000	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000
1.250000	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000

Table 3.9(b) The top-downLTM generated by ART2 networks using different vigilance values for the input pattern vectors listed in the Table 3.8.

top-down LTM (ρ=0.98)				top-down LTM (ρ=0.99)			
0.071697	3.522572	0.000000	9.358758	0.000000	3.458943	0.000000	9.382735
0.487253	4.334965	3.540942	8.272388	0.000000	4.123555	3.715506	8.318131
3.698423	4.497518	0.000000	8.129822	3.037617	5.084273	0.048271	8.057340
0.000000	6.879451	0.000000	7.257627	0.000000	6.666667	3.333333	6.666667
3.190798	5.892286	3.190798	6.702133	3.636764	3.636764	3.636764	7.766714
0.000000	0.000000	0.000000	0.000000	0.000000	6.879451	0.000000	7.257627
0.000000	0.000000	0.000000	0.000000	3.090680	6.181360	3.090680	6.533467
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Table 3.10 The data sets and the categorization results of the second ART2 example.

set no.	input pattern vectors					categories (ρ=0.98)	categories (ρ=0.99)
1	1	0	0	0	0	0	0
2	2	0	0	0	0	0	0
3	3	0	0	0	0	0	0
4	4	0	0	0	0	0	0
5	5	0	0	0	0	0	0
6	3	1	0	0	0	1	1
7	3	2	0	0	0	1	2
8	3	3	0	0	0	1	2
9	3	4	0	0	0	2	2
10	3	5	0	0	0	2	2
11	3	3	1	0	0	3	3
12	3	3	2	0	0	3	3
13	3	3	3	0	0	3	4
14	3	3	4	0	0	4	4
15	3	3	5	0	0	4	4
16	3	3	6	0	0	4	5
17	3	3	7	0	0	4	5
18	3	3	8	0	0	4	5
19	3	3	3	1	0	3	6
20	3	3	3	2	0	5	6
21	3	3	3	3	0	5	6
22	3	3	3	4	0	5	6
23	3	3	6	1	0	4	7
24	3	3	6	2	0	4	7
25	3	3	6	3	0	5	7
26	3	3	6	4	0	5	7
27	3	3	6	2	1	4	7
28	3	3	6	2	2	6	8
29	3	3	6	2	3	6	8
30	3	3	6	2	4	6	8
31	3	3	6	2	5	6	8
32	3	3	6	2	6	6	8

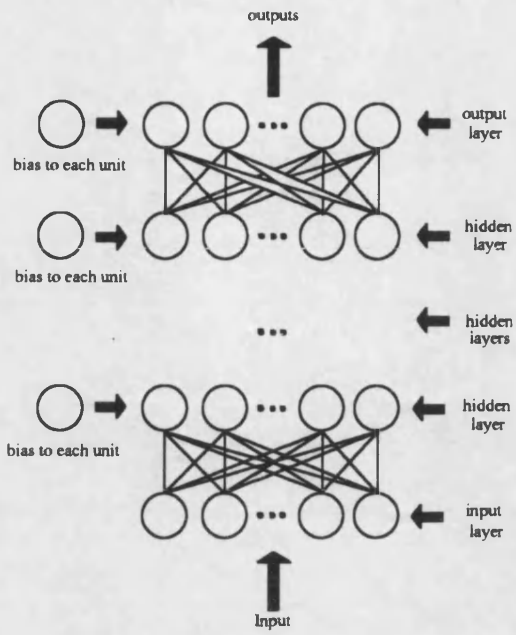


Figure 3.1 A schematic diagram of the multi-layer perceptron architecture for the BP training algorithm.

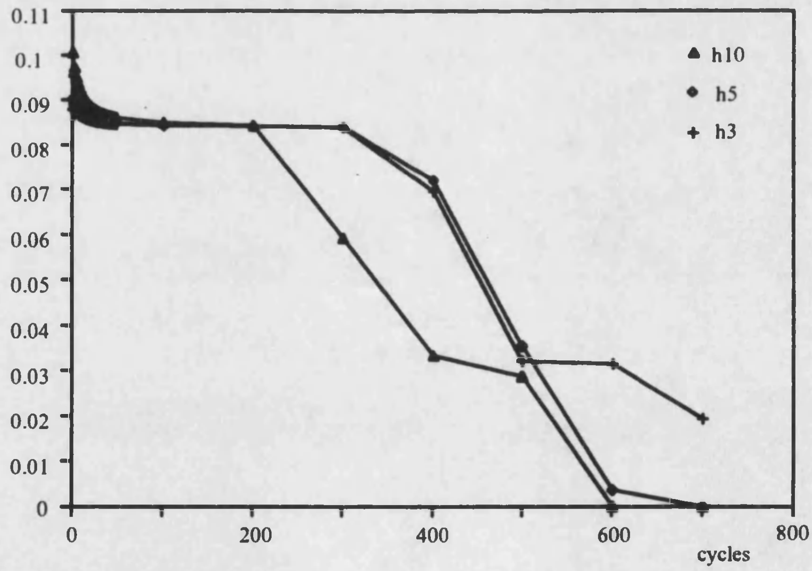


Figure 3.2 Training errors for BP networks with different numbers of units in hidden layer for the parity example.

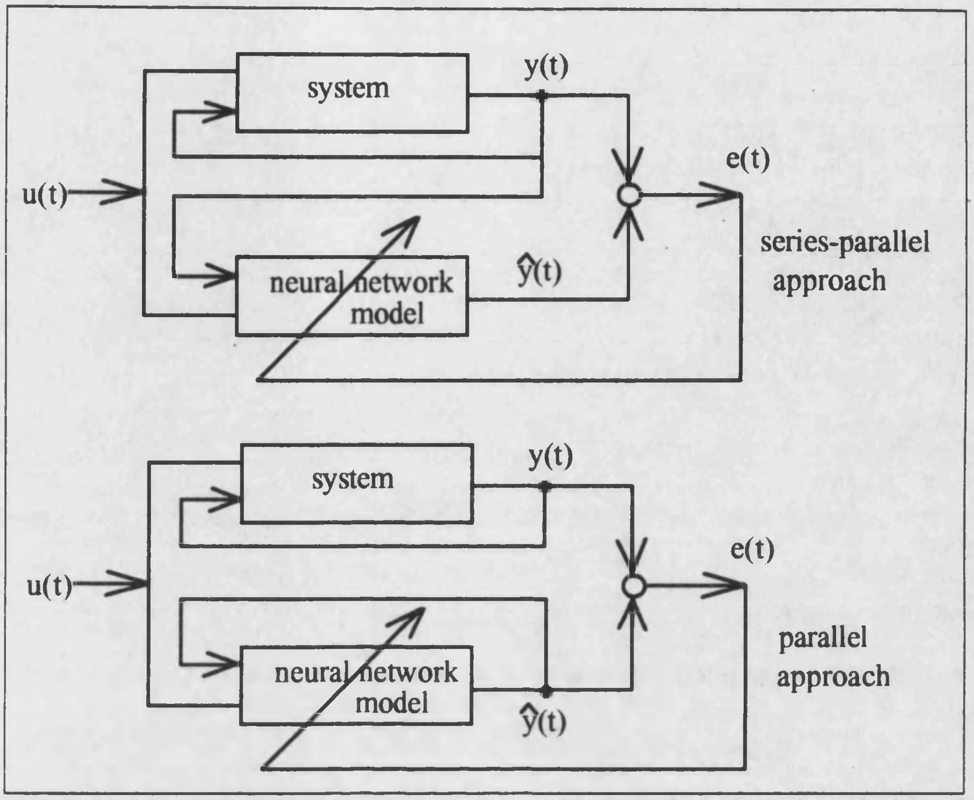


Figure 3.3 Two identification approaches using artificial neural networks.

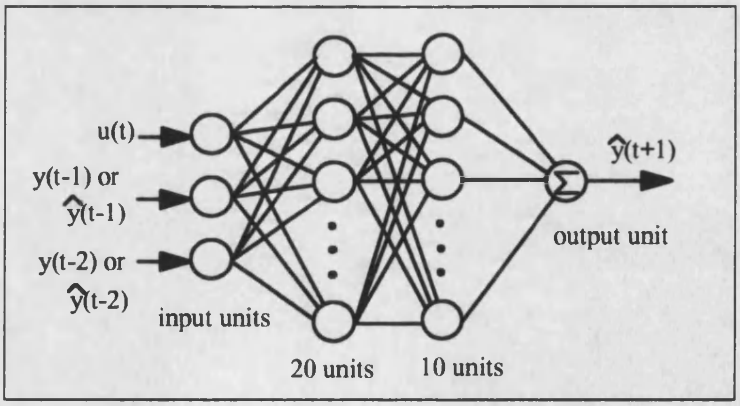


Figure 3.4 The architecture used for the identification example.

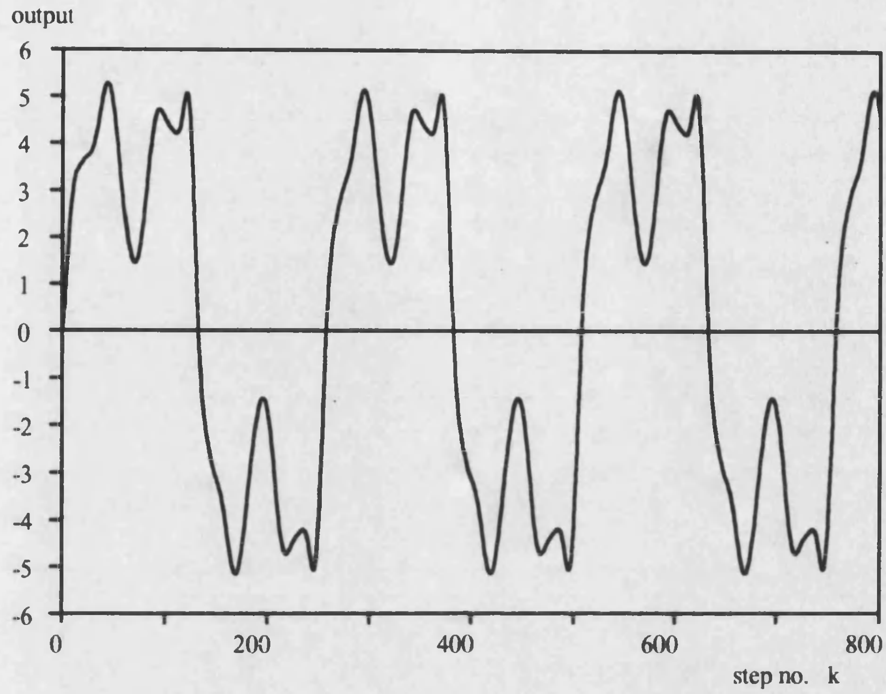


Figure 3.5 Outputs from both the system and the neural network model using the series-parallel approach.

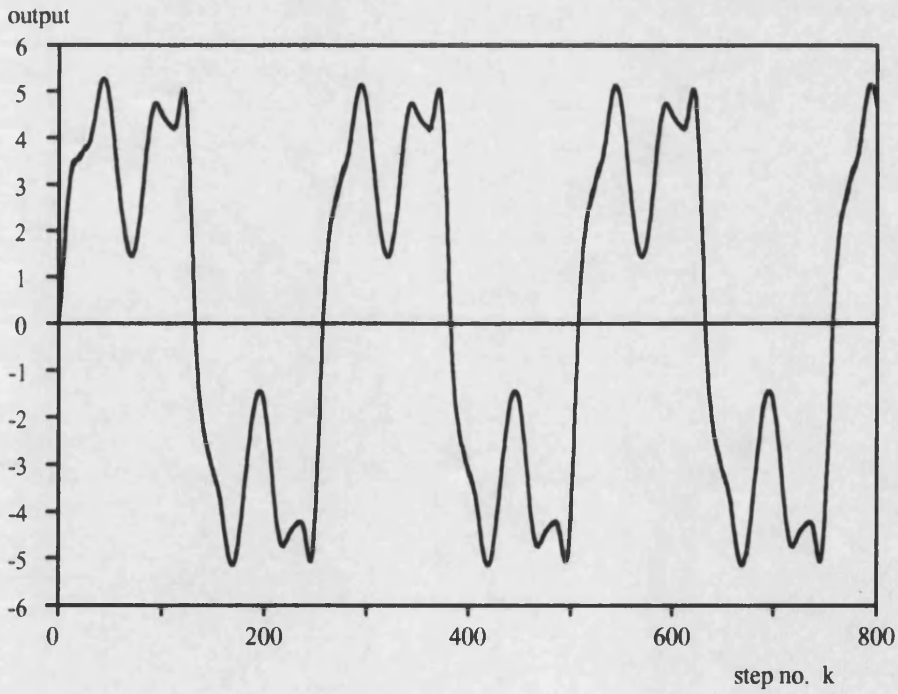


Figure 3.6 Outputs from both the system and the neural network model using the parallel approach.

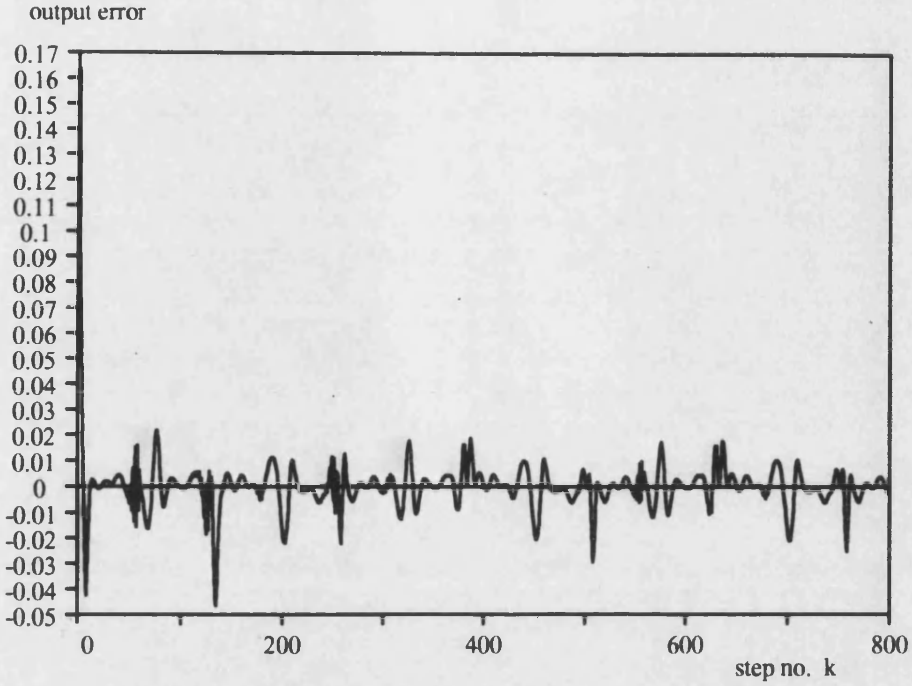


Figure 3.7 A plot of the output errors of the trained neural network model using the series-parallel approach.

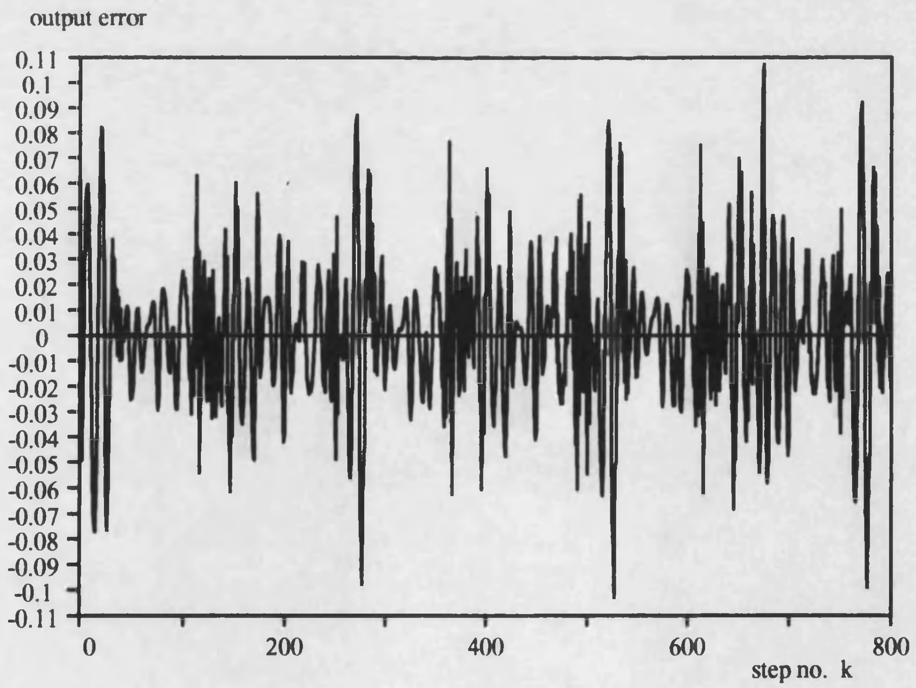


Figure 3.8 A plot of the output errors of the trained neural network model using the parallel identification approach.

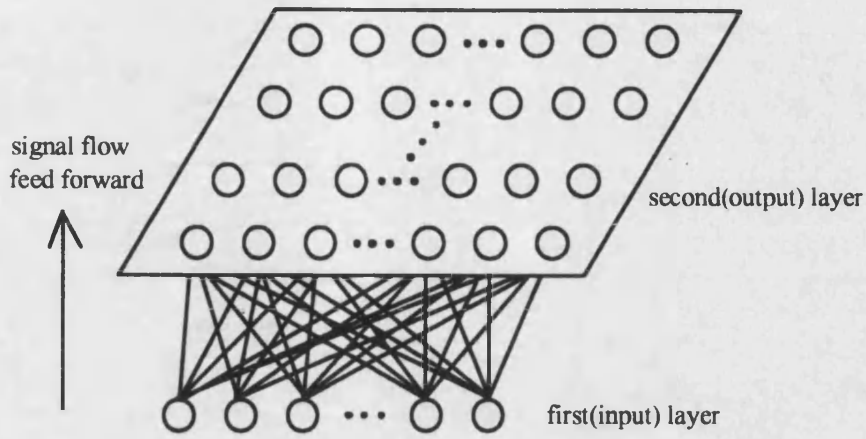


Figure 3.9 The schematic diagram of the architecture of the self-organizing map.

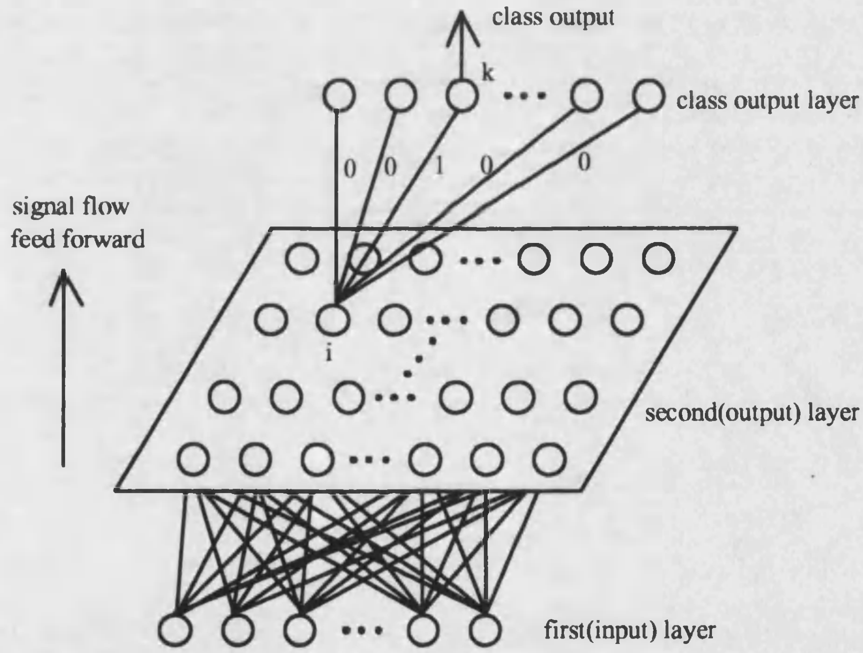


Figure 3.10 This diagram shows an additional layer added to the original network to give a crisp output of the class of input pattern vector (not all classes shown in this figure).

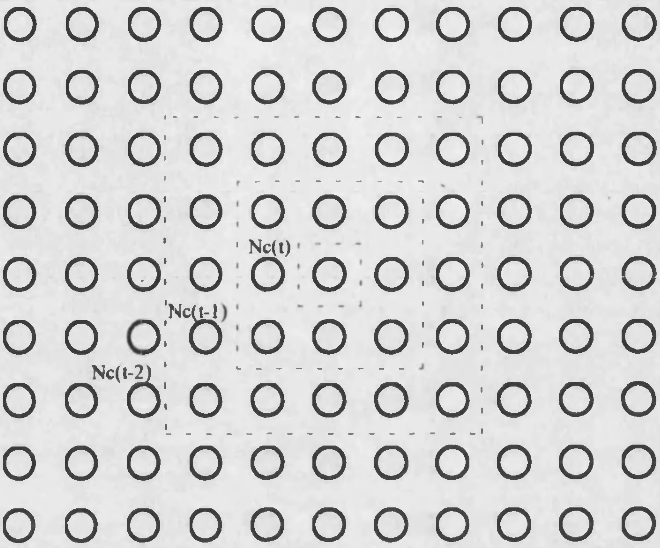


Figure 3.11 The updating neighbourhood is shrinking with time.

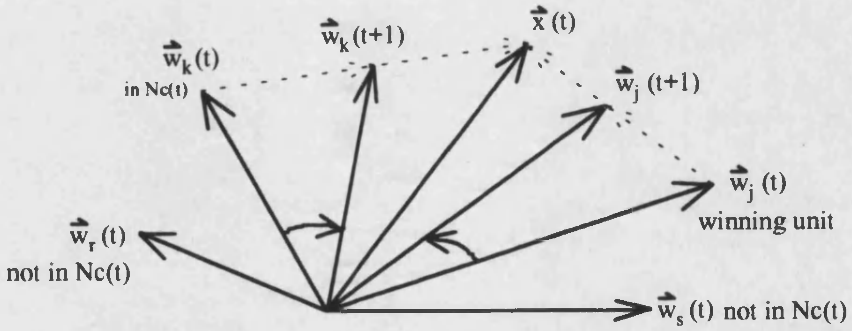


Figure 3.12 The physical meaning of the learning equation (3.12) and (3.13).

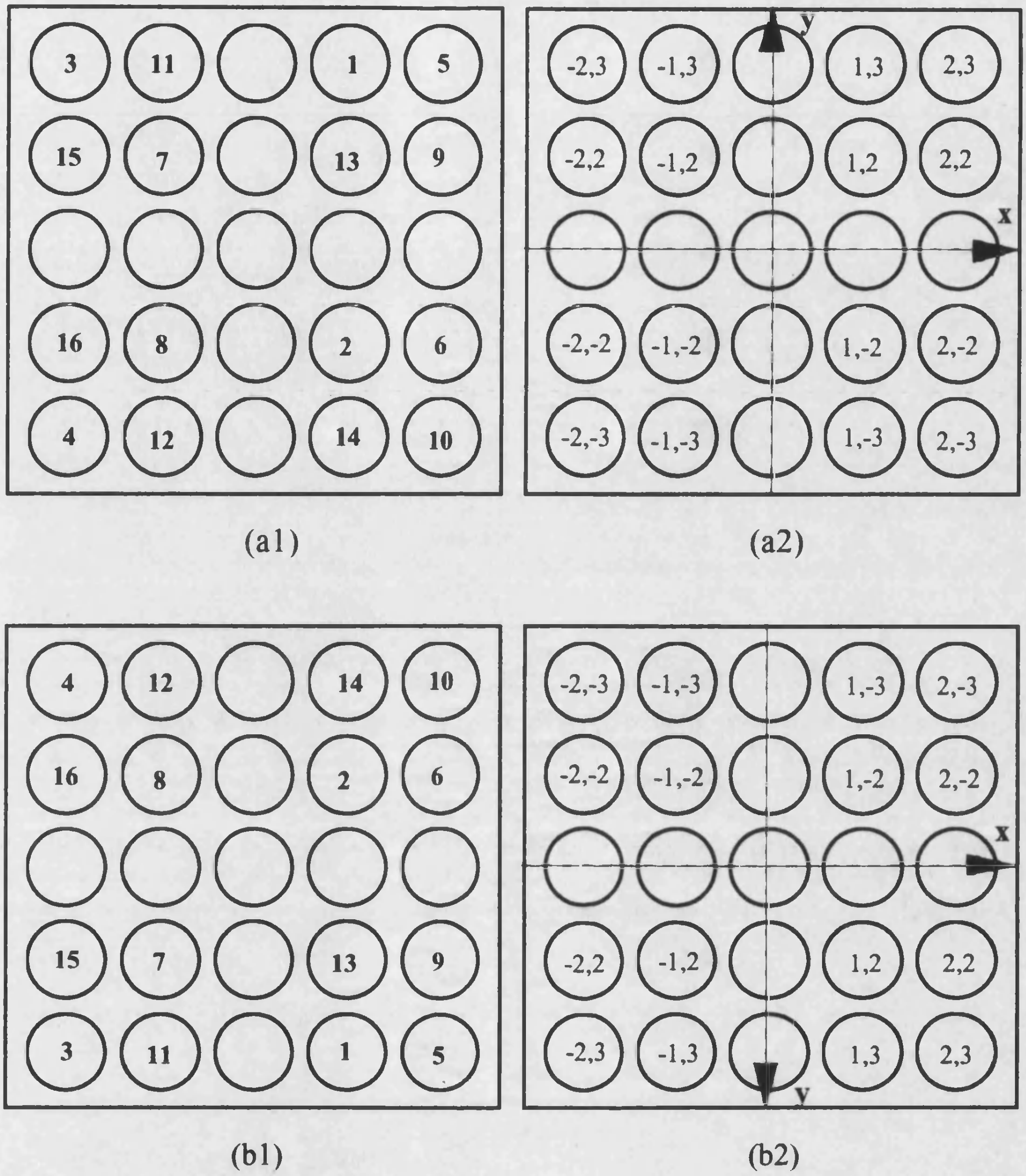
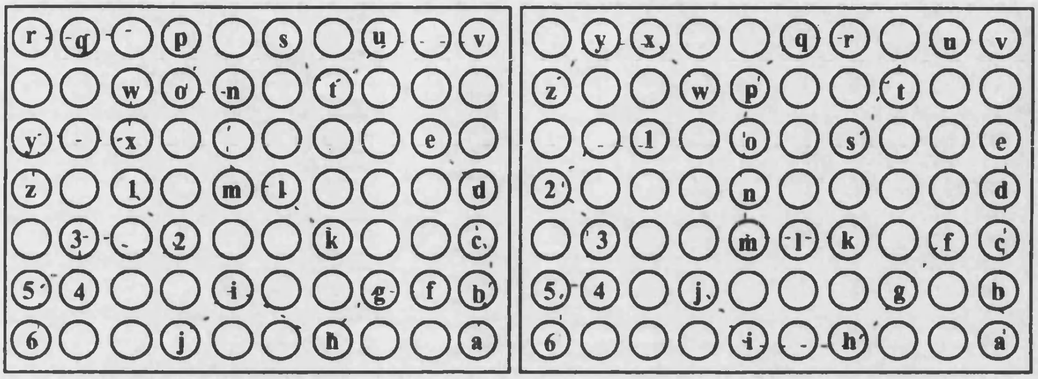


Figure 3.13 The learned map outputs. The circles stand for the processing units of the output layer of the self-organizing map. The numbers in (a1) and (b1) represent the exemplar data set numbers and the set numbers in (a2) and (b2) are the values of the exemplar data sets which actually are coordinates of points in the x-y plane.



(a) (b)

Figure 3.14 Two self-organizing maps are generated with different sets of initial weight vectors using the data sets in the Table 3.5.

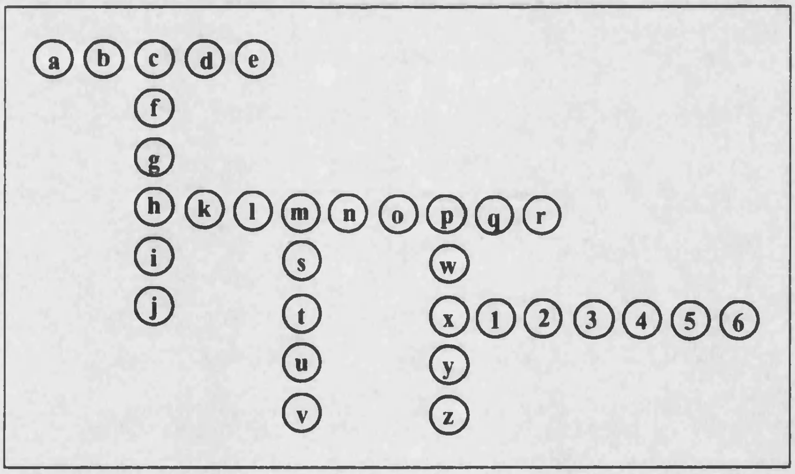


Figure 3.15 Minimal spanning tree matching to the data sets in the Table 3.5.[16]

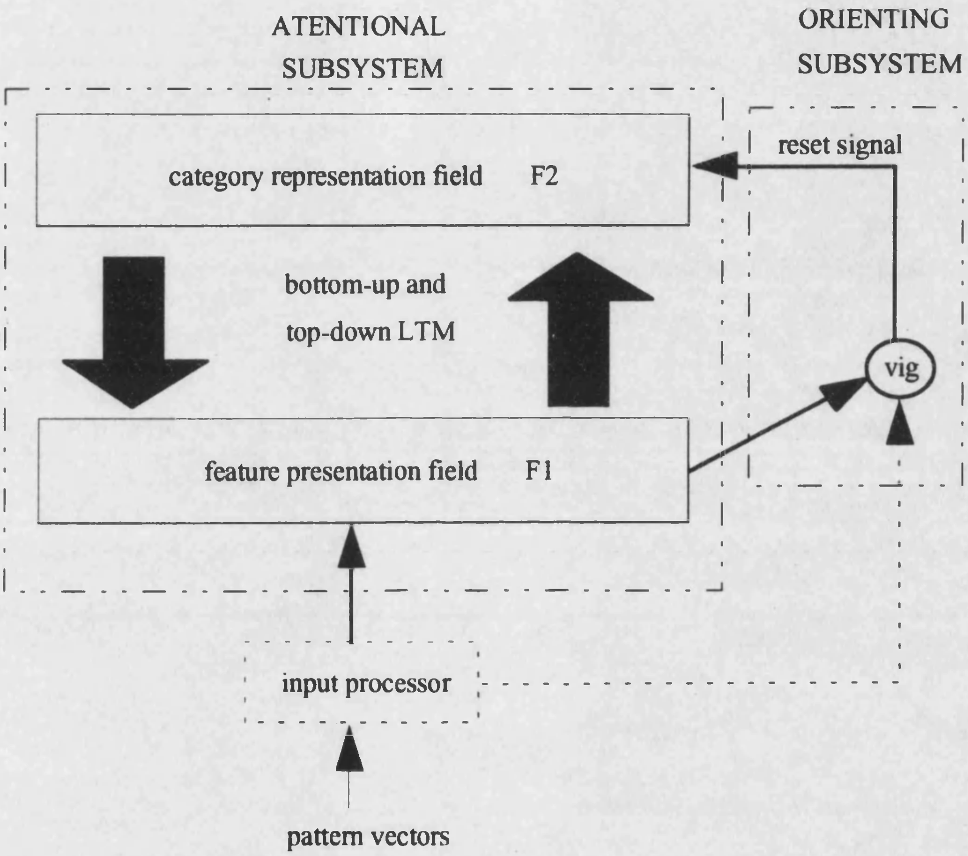


Figure 3.16 The basic architecture of the ART family.

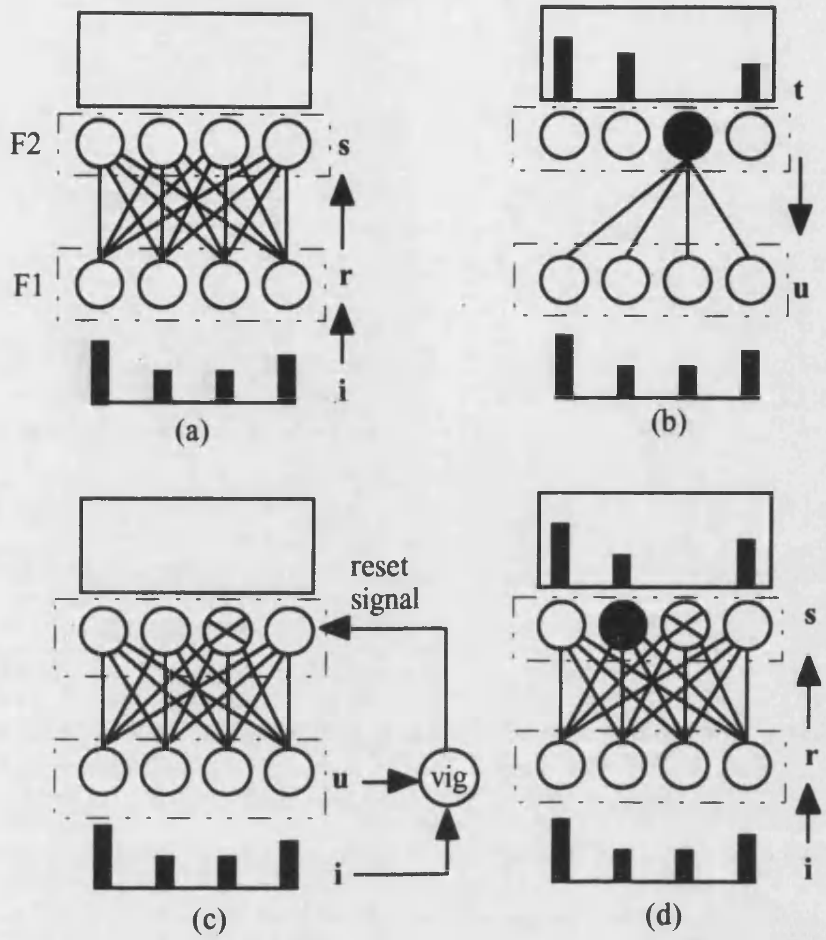


Figure 3.17 The pattern searching-matching cycles in the ART networks.

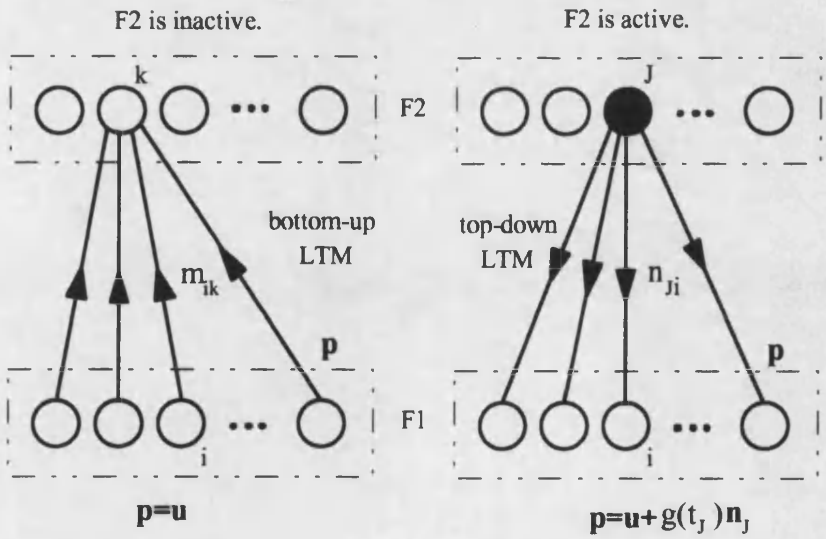


Figure 3.18 The connections between F1 and F2 fields.

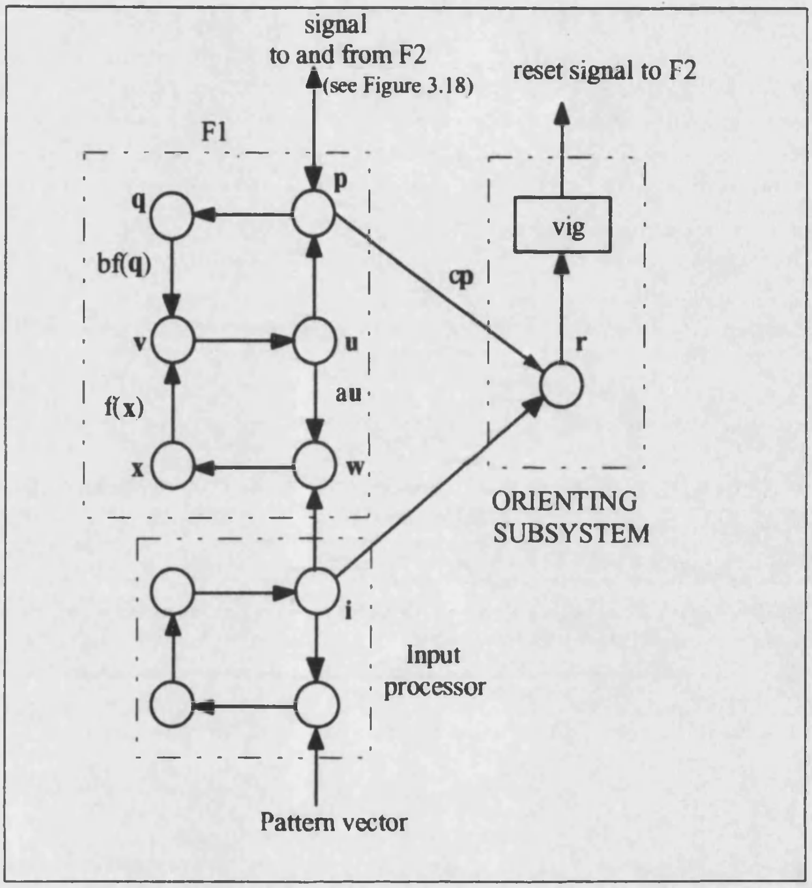


Figure 3.19 The details of the comparison layer of the ART2 network.

Chapter 4
Introduction to Fuzzy Logic and Fuzzy Neural Network

4.1 Introduction

In the previous chapter, the theories of artificial neural networks employed in this thesis for system modelling and fault classification were reviewed. In this chapter the technique derived from fuzzy logic used for system condition monitoring will be discussed.

The concept of fuzzy sets was formulated by Zadeh in 1965[80] as a means for dealing with information that was not precise. Classical crisp set theory or two-valued set theory, in contrast, only considers precise objects, which have crisp definitions such as 0 or 1, true or false, belonging to or not belonging to. The algorithm derived from fuzzy logic[81,82] is another type of reasoning method and is intended to perform imprecise modes of reasoning or approximate reasoning. The reasoning process is similar to that taken by human beings when making decision in an environment of uncertainty and imprecision. In daily life, it is common to interpret information with ambiguous meaning in order to make decisions. For example, words such as "cold", "hot", "less", "more", "tall", and so on, are ill-defined terms used almost every day. As an example, consider the following proposition and query;

If it is hot, we should wear less. He wears only one shirt. Is it hot?

It is impossible for systems based on classical two-valued logic to cope with this kind of information and related problems, but fuzzy logic systems can deal with this situation with relative ease.

Following the publication of fuzzy logic in 1968[81], this mathematical subdiscipline has found numerous applications in a variety of fields such as literature, science, finance, engineering, medicine,...etc.[83] In Japan, fuzzy logic has been used extensively to good effect. Commercial products incorporating fuzzy logic controllers such as 'fuzzy rice cookers', 'fuzzy laundry machines', 'fuzzy vacuum cleaners',...etc., can be easily found in stores[84]. The most famous application of the fuzzy logic is the automatic train operation system[85]. The current trend is to combine fuzzy logic techniques with artificial neural networks to form new systems which are more robust than systems based on either type used in isolation[86-88].

In the following sections, the basic concepts of fuzzy sets, fuzzy if-then rules, fuzzy inference, and fuzzy neural networks will be briefly reviewed. Finally a specific fuzzy neural network, which is considered to be useful in system condition monitoring will be introduced which has been programmed and tested for the future research work. The meaning of new terminologies will be explained but the rigorous definitions will not be given here and can be found in [80-81,89].

4.2 The basic concepts of fuzzy sets

4.2.1 Fuzzy set and membership functions

If C is a crisp set, which contains objects which satisfy a precise defined property, in the universe of discourse X which is a collection of objects denoted by $\{x\}$, where x is a generic element of X, then the function that defines the crisp set can be expressed as

$$\begin{aligned} h_C(x) &= 1 \quad \text{if } x \in C \\ h_C(x) &= 0 \quad \text{if } x \notin C \end{aligned} \qquad (4.1)$$

The above definition indicates that if x belongs the set C, its function value is 1; otherwise its value is zero, no matter how close is it to the boundary of the set. By contrast, the fuzzy set theory does not set a clear demarcation to judge if an element belongs to the set. Instead, it uses an imprecise but natural and intuitively plausible way to consider this problem. In fuzzy set theory, a set A is characterised by a membership function denoted by m_A. The values or grades $m_A(x)$ of the membership function measure the degree to which the object x satisfies the properties described by the fuzzy set A. The membership function that defines the fuzzy set A in the universe of discourse X can be expressed as

$$m_A : X \rightarrow [0,1] \tag{4.2}$$

where [0,1] means that the membership function only takes values between 0 and 1. The closer $m_A(x)$ is to 1, the higher is the degree or possibility of the object x satisfying the imprecisely defined property and vice versa. The general form to represent the fuzzy A can be written as

$$A=\{(x, m_A(x)) \mid x \in X\} \tag{4.3}$$

There are two other ways frequently used to express the fuzzy set A. Equation 4.4 is used to represent the fuzzy set A in the continuous universe of discourse X and equation 4.5 is for A in a discrete $U=\{x_1,x_2,...x_n\}$.

$$A = \int_X m_A(x)/x \tag{4.4}$$

$$A = \sum_{i=1}^{n} m_A(x_i)/x_i \tag{4.5}$$

Note that $\int_X$ and Σ should not be read as integration and summation; in fuzzy set theory these mean 'union' or 'or'. The symbol '/' in equations (4.4) and (4.5) simply works as a separator and $m_A(x)/x$ signifies that the grade of x is $m_A(x)$. There are an infinite number of different types of functions which can be chosen as the membership function for a fuzzy set and the final choice is not unique. The decision is usually made by experience, however, recently adaptive methods have been developed[86,87]. Finally, the support of a fuzzy set is the crisp set of all points $x \in X$ such that $m_A(x)>0$. If the support of a fuzzy set has only one element with $m_A=1.0$, then this set is known as a fuzzy singleton.

The difference between fuzzy sets and crisp sets can be clarified by a simple example. Assuming that the range of room temperatures from 17 to 21° C is considered to be comfortable. Figure 4.1 and 4.2 show the different ways the fuzzy set theory and the crisp set theory define "comfortable" for the room temperature. Now, consider the three temperature readings, 16°C, 20°C and 24°C, and the degree of comfort allocated by each system. The grades and the answers(explanations) to these temperature readings for both sets are listed in Table 4.1. It is clear that the answers from the fuzzy set are more reasonable than the answer from the crisp set. For instance, although 16°C is rather close to the boundary of the set of comfortable temperatures, the answer from the crisp set is absolutely negative; in comparison, the fuzzy set gives rational results to not just one but all temperature readings.

4.2.2 Basic operations for fuzzy sets

Let A and B be two fuzzy sets in X with membership functions $m_A(x)$ and $m_B(x)$, respectively. The basic operations for fuzzy sets defined in [80] are shown as follows:

equality(=):	$A=B$ iff $m_A(x)=m_B(x) \quad \forall x \in X$	(4.6)
containment($\subset$):	$A \subset B$ iff $m_A(x) \leq m_B(x) \quad \forall x \in X$	(4.7)
union($\cup$):	$m_{A \cup B}(x)=\max(m_A(x),m_B(x)) \quad \forall x \in X$	(4.8)
intersection($\cap$):	$m_{A \cap B}(x)=\min(m_A(x),m_B(x)) \quad \forall x \in X$	(4.9)
algebraic product:	$m_{AB}(x)=m_A(x)m_B(x) \quad \forall x \in X$	(4.10)
complement:	$m_{A^*}(x)=1-m_A(x) \quad \forall x \in X$	(4.11)

4.2.3 Linguistic variables and hedges

Linguistic variables are defined as those which can take natural words as their values. For examples, the linguistic variable error may take 'big', 'medium', 'small', 'very big' and 'very small' as its values. Each of these values is a fuzzy set and is called a label of the linguistic variable error. The modifying terms such as 'very', 'much', 'more or less'...etc. are called hedges. Some commonly used hedges are briefly defined as: if A is a fuzzy set in the universe of discourse X,

$$\text{very A:} \quad m_{\text{very A}}(x)=(m_A(x))^2 \tag{4.12}$$

$$\text{more or less A:} \quad m_{\text{more or less A}}(x)=(m_A(x))^{0.5} \tag{4.13}$$

$$\text{not A:} \quad m_{\text{not A}}(x)=1.0-(m_A(x)) \tag{4.14}$$

The fuzziness of a fuzzy set can be changed, i.e. increased or decreased, by applying these operations. The concepts of linguistic variables and the membership functions make it possible to quantitatively describe and subsequently manipulate the linguistic information.

4.3 Fuzzy reasoning(approximate reasoning)

4.3.1 Fuzzy if-then rule

The techniques derived from fuzzy logic have been very successfully applied in many engineering systems. The implementation of these techniques are usually completed by collecting operational knowledge and rules then transforming them into if-then rules. Usually most fuzzy if-then rules come from experts' experiences, although nowadays several training algorithms are also available for generating the rules[86-88]. The rules can be expressed in the following generalised form.

$$\text{If } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and ... and } x_n \text{ is } A_n^i \text{, then y is } B^i. \tag{4.15}$$

where i stands for the ith number of rules, A_i and B are fuzzy sets.

The if-part is called the condition part and the then part is called the action part. The expression in (4.15) is a general forms for multiple input and single output rules. For multiple output rules, each rule can be decomposed into several rules in the form of (4.15) with the same if-part. Rules with other conjunctions rather than "and" can be found in the literature. In applications, the collection of if-then rules are crucial to the performance of the fuzzy logic system.

4.3.2 Fuzzy relation and fuzzy implication

Let X and Y be two universes of discourse. A fuzzy binary relation R is the fuzzy set in the product space X×Y, that is mapping from X→Y, and has the membership function $m_R(x,y)$, where $x \in X$ and $y \in Y$. A generalisation to the n-ary relations is straight forward.

Each of the if-then rules mentioned in the last section are regarded as a relation from A→B, where A stands for the if-part ;

$$\text{If } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and ... and } x_n \text{ is } A_n^i \tag{4.16}$$

and B represents the then-part

$$\text{then y is } B^i \tag{4.17}$$

The expression A→B is understood as a fuzzy implication. The so-called 'implication rules' are employed for explaining the meaning, in numerical form, of the fuzzy implications. The most popular implication rules found in the literature are the mini-operation rule and the product-operation rule. Both of them are shown as follows.

$$\text{mini-operation rule} \quad R_{min}=\int_{X\times Y} m_A(x) \wedge m_B(y)/(x,y). \tag{4.18}$$

product-operation rule $R_{product}=\int_{X\times Y} m_A(x)m_B(y)/(x,y)$. (4.19)

Before moving to the next section, there is a problem that needs to be solved. If the rules are for multiple inputs with conjunctive 'and', we have to know how to manipulate the fuzzy sets related by the conjunctive 'and'. For example, let the if-part be If x_1 is A_1 and x_2 is A_2 and ... and x_n is A_n , and also the membership functions for the fuzzy sets be $m_{A_1}(x_1), m_{A_2}(x_2), \ldots, m_{A_n}(x_n)$. Two operations are often applied to combine these grades of membership functions and are shown as follows.

$$m_A(\mathbf{x}) = m_{A_1\times\cdots\times A_n}(\mathbf{x}) = \wedge(m_{A_1}(x_1), m_{A_2}(x_2), \ldots, m_{A_n}(x_n)) \quad (4.20)$$

$$m_A(\mathbf{x}) = m_{A_1\times\cdots\times A_n}(\mathbf{x}) = m_{A_1}(x_1)m_{A_2}(x_2)\ldots m_{A_n}(x_n) \quad (4.21)$$

In the equation (4.20) the fuzzy intersection operation is used and in the equation (4.21) the algebraic product is employed. Thereafter, when an if-then rule is considered as an implication, for instance A→B, the grade of fuzzy sets A can be obtained from one of the above equations.

4.3.3 Compositional rules of inference

Consider the single rule:

If x_1 is A_1 and x_2 is A_2, then y is B. (4.22)

where A_1, A_2 are fuzzy sets.

However, if the inputs are

x_1 is A_1' and x_2 is A_2' (4.23)

We would expect the output inferred from the above rule to be y is B'. The process of inference may be represented as the operation

$$B' = (A_1', A_2') \otimes (A_1 \text{ and } A_2 \rightarrow B)$$
or in the membership form (4.24)
$$m_{B'} = (m_{A_1'}, m_{A_2'}) \otimes (m_{A_1 \times A_2} \rightarrow m_B)$$

The algorithms by which the above '$\otimes$' operation is carried out is called the compositional rule of inference. Different compositional rules have been suggested which differ in their results and their mathematical properties. The most popular form of compositional rule is the sup-* rule, where 'sup' means the least upper bound and '*' denotes an operator such as product, min, etc. The most commonly used composition rules are sup-min and sup-product rules[82,89]. The sup-min and sup-product rules will give the same results if the inputs are all fuzzy singletons. If the then-part of an if-then rule is singleton then the min and product implication rules will have the same effect on the inferences.

4.3.4 Summary of the fuzzy reasoning process

Many different inferential algorithms have been proposed based on different combinations of the sup-star composition and fuzzy implications. Of course, the results from different inferential algorithms will generate different results. The steps of fuzzy reasoning, using the combination of sup-min or sup-product compositional rule and min or product implication rules, are summarised as follows:

(a) Apply fuzzy inputs to the if-part of each of the if-then rules using a compositional rule.

(b) Calculate the firing strength w_i(shown in Figures 4.3 and 4.4) of each of the rules by an implication rule.

(c) Combine the firing strength of rules to acquire the final inference and/or apply a defuzzification technique to acquire a crisp value.

The steps listed above are shown in Figure 4.3 in which the sup-min compositional rule and min implication rule are employed. In Figure 4.4 the sup-min compositional rule and production implication rule are used. The defuzzification technique shown in the third step still needs further explanation.

The defuzzification process is the inverse mapping from a fuzzy set to obtain a crisp point in the universe of discourse in which the fuzzy set is defined. For example, in Figures 4.3 and 4.4 the final fuzzy outputs are shown in the shaded shapes, but the results can not directly be applied to the real systems in which the fuzzy logic systems are used. We must transform the fuzzy outputs to a crisp number, such as y^0 shown in the figures, in order to be used in the real systems. For example, in a control system the y^0 could be a control command to the system. A variety of techniques of defuzzification can be found in the literature[84-86,89] and there is no definite criterion for choosing the most appropriate technique. The choice of defuzzification technique is system dependent and will influence the performance of the system. In general, the centre of average defuzzification method is often used. This is expressed as

$$y^0 = \frac{\sum_{i=1}^{n} y^i m_{B'^i}(y^i)}{\sum_{i=1}^{n} m_{B'^i}(y^i)} \tag{4.25}$$

where y^i is the support value at which the membership function $m_{B^i}(y^i)$ achieves its maximum value, $m_{B'^i}(y^i)$ is given by the membership in the action part of each if-then rule, and n is the numbers of if-then rules.

4.4 Introduction to the fuzzy neural network

4.4.1 The relationship between fuzzy logic and artificial neural networks

The techniques derived from artificial neural networks and fuzzy logic have been extensively and successfully applied to areas such as pattern recognition and control. Recently, it has become increasingly popular to use artificial neural networks and fuzzy logic in combination. The marriage of these two different techniques are quite natural. Firstly, they both aim to mimic the remarkable ability of the human mind to learn and make rational decisions in circumstances of uncertainty and imprecision. Secondly, each of the two methods is good at dealing with different types of problems. While artificial neural network methods are suitable for coping with sensor data used in function representation and pattern recognition, the fuzzy logic methods are specially designed for handling data and information that possesses nonstatistical uncertainty. In combination, the two methods can complement each other. Bezdek[90] stated that existing efforts at merging these two methods may be characterised as (1) fuzzification of conventional computational neural network architectures and models and (2) the use of computational neural networks as tools in fuzzy models.

In the literature, fuzzy neural networks can be roughly categorised into three groups. These are

(1) Fuzzy neural networks are derived from BP networks mentioned in the last chapter. This group of fuzzy neural networks can be further divided into two sub-groups. The first sub-group makes use of the membership concepts in the input and output data but keeps the original BP neural network structure and algorithm[91]. In this sub-group the only major difference with the original BP network is that the input data and/or output data are transferred into the grade of the membership functions and the mechanisms of fuzzy inference are not included. The second sub-group employs the basic ideas of the algorithms and structures of BP networks. However, the mechanisms of fuzzy inference are embodied in these neural networks. Most of the

fuzzy neural networks are categorised in this sub-group[86-88,92,93]. Due to the coverage of the fuzzy inference, the structures and the learning algorithms of these fuzzy neural networks need different degrees of modification.

(2) The second group of fuzzy neural networks are those derived from other artificial neural networks rather than BP networks. The structures and the learning algorithms of this group of networks differ completely from those of the first group. For examples, fuzzy Kohonen clustering networks are based on Kohonen's self-organising maps[94], while the fuzzy ART[95] and fuzzy ARTMAP[96] are both derived from adaptive resonance theories. Fuzzy min-max neural networks are based on their inventor's own ideas[97,98].

(3) The third group of fuzzy neural networks can be called computational neural-like fuzzy networks. The networks in this group employ network structures similar to multi-perceptron networks to implement the fuzzy inference and no weight adjustments are needed by BP algorithm[99,100]. The processing units in this type of network are usually assigned to perform different fuzzy operations and the values of the connecting weights between units are assigned during the building of the network structures.

4.4.2 An example of the fuzzy neural networks

In this research, a special fuzzy neural network was investigated to investigating the possibility of applying this newly developed network for future research in condition monitoring. This network was first proposed by Simpson[97] and was used for classification purpose. It is called the fuzzy min-max neural network. This neural network employs a supervised learning scheme for encoding, but a slightly different scheme can be used for unsupervised learning[98]. The structure of this neural network is composed of three layers of processing units, shown in Figure 4.5. The first layer is the input layer, the second layer is the hyperbox layer and the third layer is the class output layer. The main idea of this neural network is try to build hyperboxes for different groups of patterns and, in the two dimensional case, the hyperboxes degenerate to rectangles. Each unit in the second layer represents a hyperbox which stands for a group of patterns, and one or several hyperboxes are used to stand for a class of patterns, which in turn is outputted from each of the units in the last layer. The hyperboxes are encoded in the weights connecting the input units to the hyperbox units in the second layer and are represented by $\mathbf{w}_j$ and $\mathbf{v}_j$

$$\mathbf{w}_j = (w_{1j}, w_{2j}, \cdots, w_{nj}) \tag{4.26}$$

$$\mathbf{v}_j = (v_{1j}, v_{2j}, \cdots, v_{nj}) \tag{4.27}$$

where $\mathbf{w}_j$ and $\mathbf{v}_j$ are understood as the maximum point and the minimum point in the jth hyperbox. They are also the two sets of connecting weights from the ith input units to the jth hyperbox unit, shown in Figure 4.6. The outputs from the hyperbox units in the second layer are the grades of the hyperbox membership which is defined as

$$m_j(\mathbf{x}^p) = \frac{1}{2n}\sum_{i=1}^{n}[\vee(0,1-\vee(0,a(\wedge(1,x_i^p - w_{ij})))) + \vee(0,1-\vee(0,a(\wedge(1,v_{ij} - x_i^p))))] \tag{4.28}$$

where $m_j(\mathbf{x}^p)$ is the membership function of the input pattern $\mathbf{x}^p = (x_1^p, x_2^p, \cdots, x_n^p)$.

j is used to denote the jth unit in the second layer.

p stands for the pth input pattern.

n is the number of dimensions of the input patterns.

$\vee$ and $\wedge$ represent the fuzzy union and intersection, respectively.

a is the sensitivity parameter which regulates how fast the grades of the membership function decrease as the distance between the pattern $\mathbf{x}^p$ and a hyperbox, in other words it is responsible for the shape of the membership function.

The values of the weights between the second and the third layers are either 1 or 0. If the compressed hyperbox represented by the jth hyperbox unit belongs to the kth class of patterns, then the value of the weight h_{ik} connected from ith hyperbox unit to the kth class output unit in the last layer is set to 1, otherwise it is set to 0. Each of the class output units performs a fuzzy union operation after receiving the input $m_i h_{ik}$ and subsequently generates an output $\vee(m_1 h_{1k}, m_2 h_{2k}, \cdots, m_n h_{nk})$, where $\vee$ represents the fuzzy union operator and n is the number of the hyperbox unit. When the neural network is used for the task of classification, and if a hard decision is needed, the class represented by the class unit in the last layer with the max value of $m_i h_{ik}$ is considered to be the class of the input pattern.

The fuzzy min-max neural network can also automatically, easily and very quickly adopt new patterns to the coded neural network without causing the stability-plasticity problem. It is one of the most promising neural networks for applications in classification. The neural network adopt a newly encountered pattern by either expanding the size of one of the hyperboxes previously encoded or by adding a new hyperbox to the uncommitted hyperbox unit, according to the following criterion

$$n \cdot b \geq \sum_{i=1}^{n} (\vee(x_i^p, w_{ij}) - \wedge(x_i^p, v_{ij})) \tag{4.29}$$

where n is the dimensions of the input patterns.

b is a parameter regulating the size of a hyperbox.

If an existing hyperbox is in the same class as the input pattern and this criterion is satisfied as well, then the size of the existing hyperbox can be expanded to include the new pattern. If there is no existing hyperbox satisfying the criterion then the new exemplar pattern is used to establish a new hyperbox in one of the uncommitted hyperbox units. In addition, the connections between the units in the second layer and the units in the last layer are changed accordingly. However, this expanding process can cause overlap between hyperboxes which do not belong to the same class. Thus the overlap of hyperboxes must be eliminated and the process of elimination is called hyperbox contraction. The contraction of the hyperboxes is considered with four different cases, the details of which can be found in the reference[97]. After the contraction process, all overlaps between hyperboxes in different classes will be eliminated.

The fuzzy min-max neural network was programmed in 'C' and tested with data used in the original paper.[97] The test data sets are shown in Table 4.2 and the test results are shown in Figures 4.7(a) and (b), and Figure 4.8 for different values of parameter b. The separation of the w_{ij} and v_{ij} in Figure 4.7(a) and (b) is for clarity. In Figures 4.7 and 4.8, the initial weights connected to the uncommitted hyperbox units, which are second layer units, are not shown and the input pattern vector with the final output values are associated with the last set of test data. The effects of the parameters can be understood by examining Figures 4.7 to 4.10. The data sets are encoded in two different hyperboxes, each of them represents a different class, when the parameter b is set to 0.3 (Figure 4.7 and Figure 4.9). However, when this parameter is smaller(b=0.2), the four data sets are assigned to four different hyperbox units, shown in Figure 4.8 and Figure 4.10. It is noted that because the parameter b is small, four different hyperboxes shown in Figure 4.10 actually degenerates to four points. Also, in Figure 4.7 we can see from the final outputs that after the contraction process a pattern which is very close to the boundaries of the hyperboxes could be misclassified, if a hard decision is made. However, when the parameter b is smaller, it is still possible for the network to give the correct classification as shown in Figure 4.8. According to this observation, if the network is to be used for fault classification, the value of parameter b must be carefully chosen and also a soft decision scheme may be needed instead of a hard one. Further research of this potential fuzzy network for condition monitoring is still needed.

4.5 Closure

People engaged in system condition monitoring frequently need to deal with imprecise information and to make decisions according to this information. This is the reason why many monitoring systems have been knowledge based in the past[13,27,30]. The techniques derived from fuzzy logic are especially suitable for dealing with problems in which information is ambiguous and human decision making is heavily relied on. In this chapter the basic concepts of fuzzy reasoning were introduced and this technique will be tested for condition monitoring described in the next chapter. Fuzzy neural networks, which combine fuzzy reasoning with artificial neural networks, are newly developed topics in literature. Fuzzy neural networks can be very useful techniques for condition monitoring and this needs more research. In this chapter, a computer program for a fuzzy neural network, called the fuzzy min-max neural network, was coded and an example was shown using this program. This test example and the underlining theory of this fuzzy neural network showed the possibility of using this fuzzy neural network for condition monitoring. A program for simulating a neural-like computing network, which applied the theories discussed in this chapter, was also coded. This network has been used in this research and the results will be shown in the next chapter.

Table 4.1 Comparison between the crisp and fuzzy sets for comfortable temperature.

temperature	crisp set		fuzzy set	
°C	grade	answer	grade	answer
16	0	uncomfortable	0.88	degree of comfortable=0.88
20	1.0	comfortable	1.0	degree of comfortable =1.0
24	0	uncomfortable	0.57	degree of comfortable = 0.57

Table 4.2 Test data sets for the example.

input pattern vector	pattern class
(0.5, 0.5)	1
(0.6, 0.6)	2
(0.2, 0.2)	1
(0.4, 0.3)	2

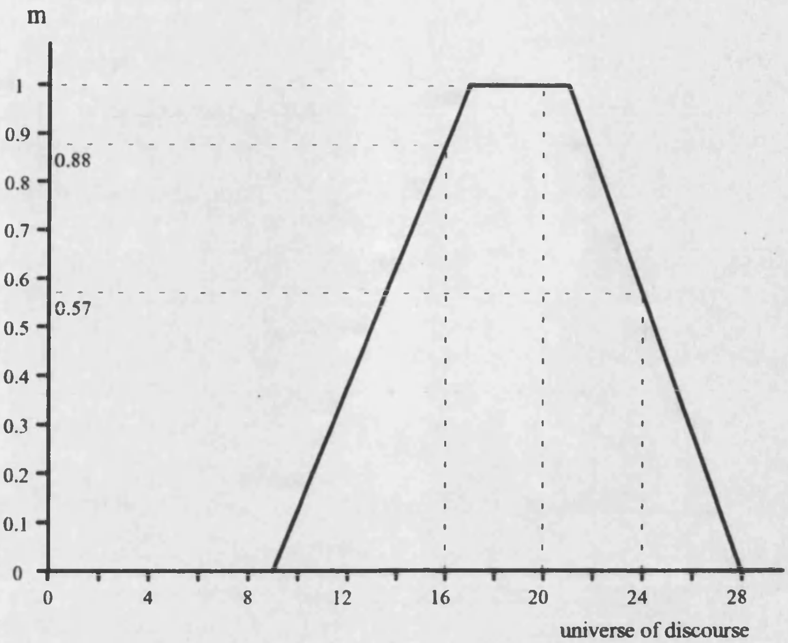


Figure 4.1 The fuzzy set of comfortable temperature.

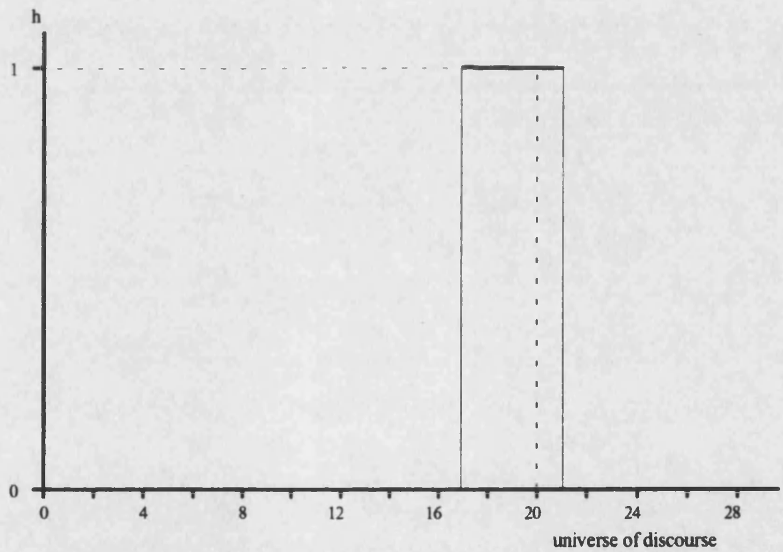


Figure 4.2 The crisp set of comfortable temperature.

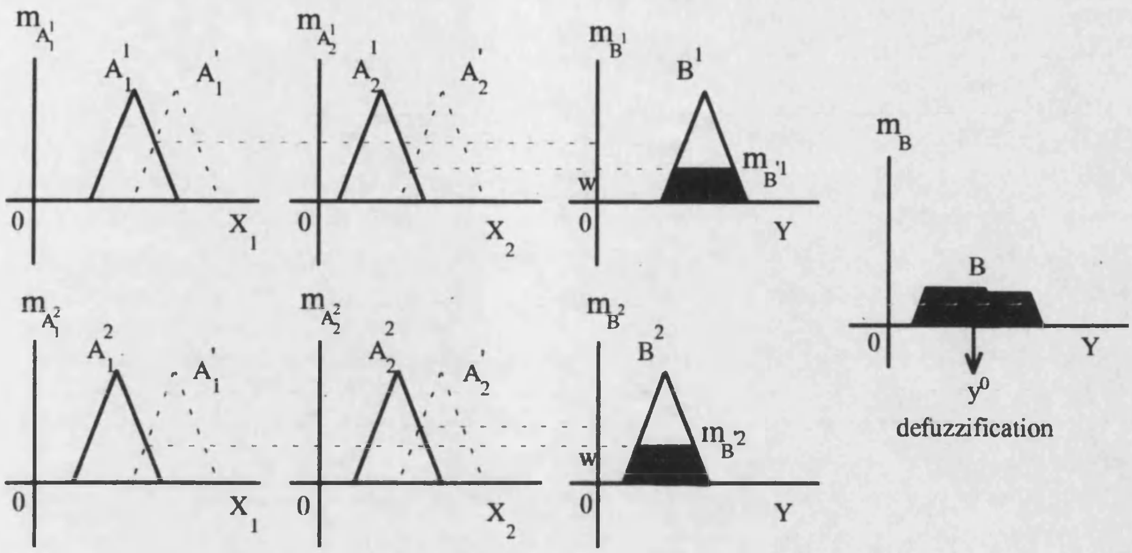


Figure 4.3 Graphical representation of the sup-min composition and min implication.

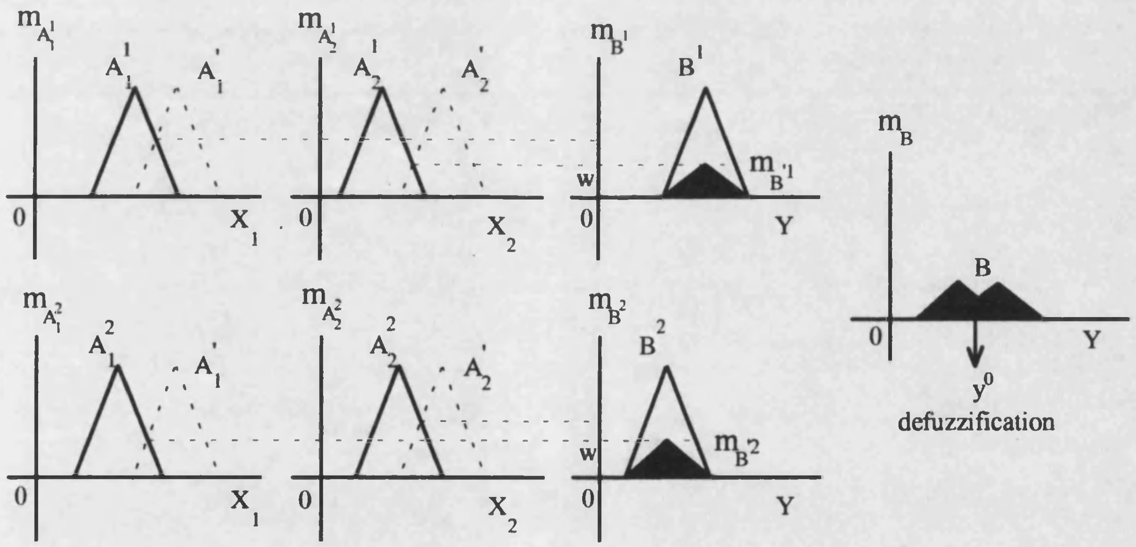


Figure 4.4 Graphical representation of the sup-min composition and product implication.

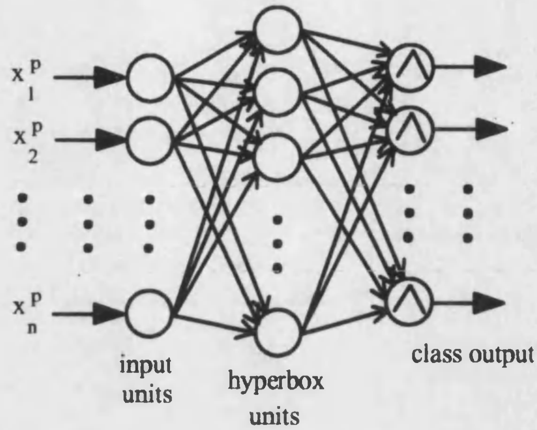


Figure 4.5 The structure of the fuzzy max-min neural network.

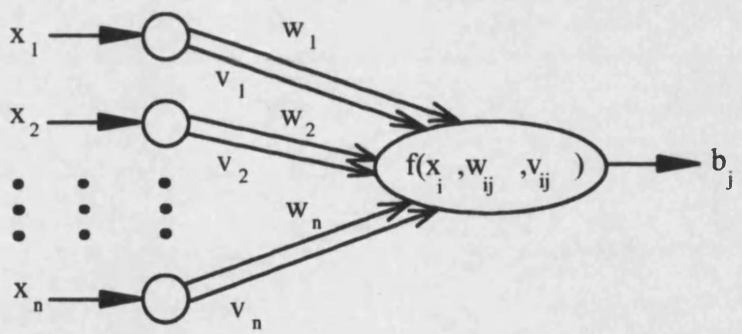


Figure 4.6 Diagram shows the connecting weights between the input units and the one of hyperbox units.

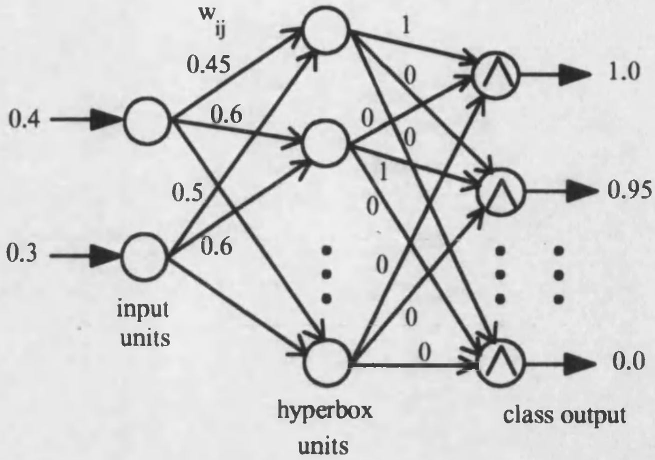


Figure 4.7(a) Diagram fro showing w_{ij} in the test results of the example using the fuzzy min-max neural network(a=4, b=0.3).

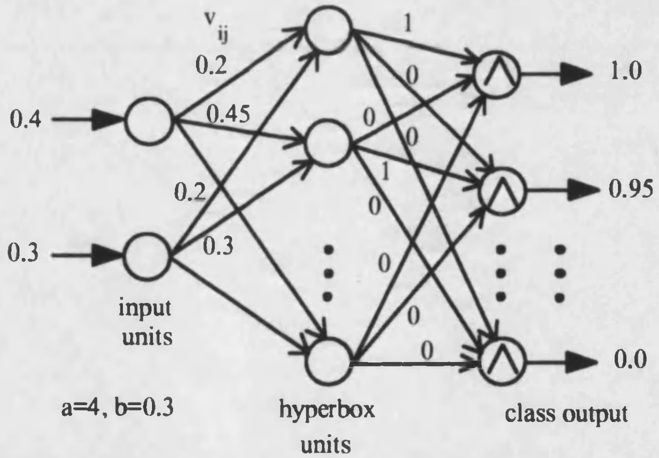


Figure 4.7(b) Diagram fro showing v_{ij} in the test results of the example using the fuzzy min-max neural network(a=4, b=0.3).

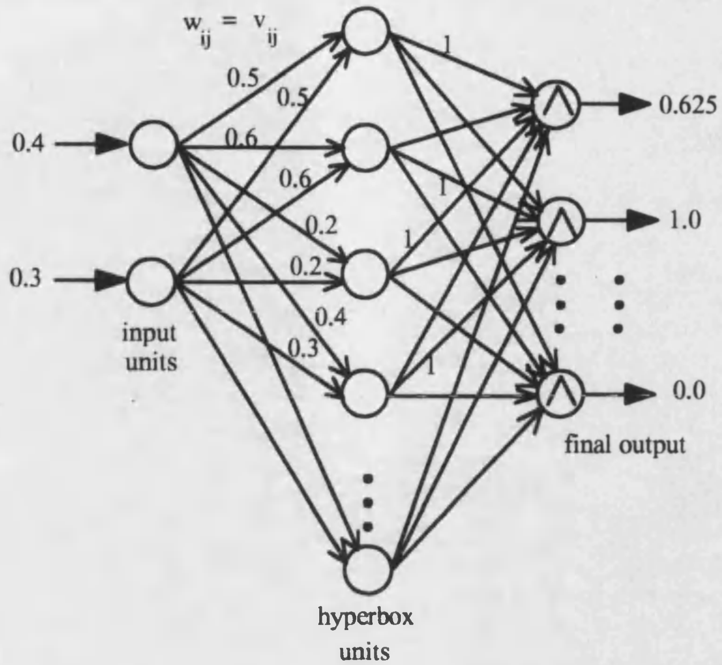


Figure 4.8 Diagram for showing the test results of the example using the fuzzy min-max neural network (a=5, b=0.2).

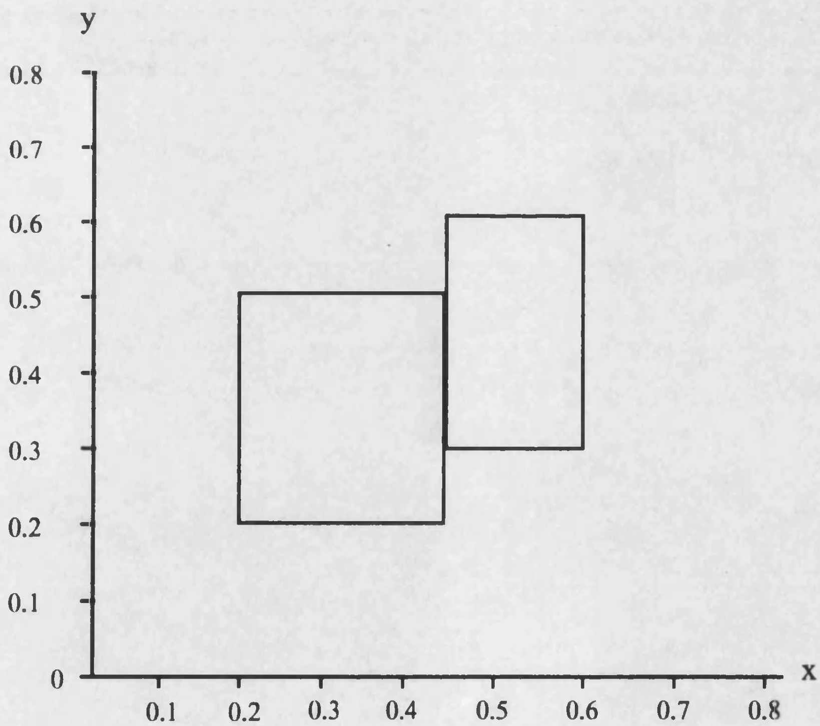


Figure 4.9 Hyperboxes generated(a=4,b=0.3) using data in Table 4.2.

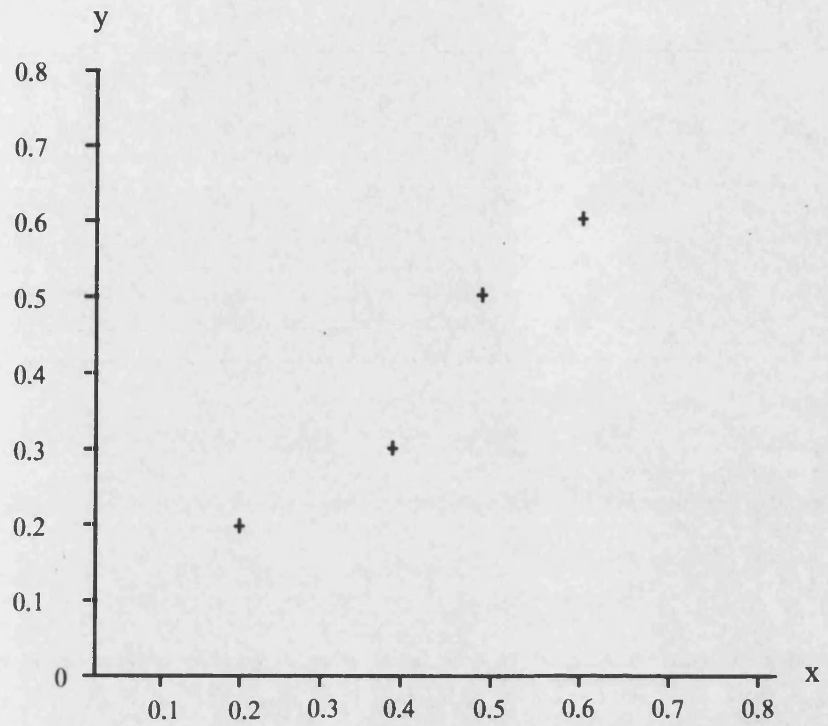


Figure 4.10 Hyperboxes generated(a=5,b=0.2) using data in Table 4.2.

Chapter 5
Simulation and Experimental Results

5.1 Introduction

In this chapter condition monitoring techniques based on artificial neural networks, derived from fuzzy logic, are introduced and tested by means of simulation and experimental data. The methods used for the modelling of fluid power systems using artificial neural networks are demonstrated firstly.

An accurate model which represents the monitored healthy system is an essential part of the monitoring process and is critical to the success of the monitoring tasks. Suppose that there is a change in one of the many outputs of the monitored system at a certain time. The difference between two output signals, one from the monitored system and the other one from the reference model, will generate an error signal and subsequently trigger an alarm signal. If an accurate reference model is available then the correct outputs of the monitored system, responding to the input commands, can be predicted with confidence and as a result the chance of causing a false alarm can be reduced to a minimum. The second central issue relating to the task of monitoring is to classify the types of detected faults. Eventually it is anticipated that the locations of the faulty elements in the monitored system can be pinpointed according to the features or signatures of the error signal. Two different approaches for classifying the system faults will be demonstrated. The first approach makes use of the pattern recognition technique and the second approach applies fuzzy logic, but in both approaches artificial neural networks are still playing very crucial roles. The basic concept of pattern recognition is to compare an unknown pattern vector with the labelled or recognised pattern vectors using some sort of similarity measuring method[101] in order to categorise the unknown pattern to an existing class. Not are those traditional pattern recognition techniques but are the methods based on artificial neural networks covered in the following sections. Fuzzy logic introduced in the last chapter will also be combined with the computational neural-like networks for the task of condition monitoring. All the simulation and experimental results will be presented in sections 5.2 to 5.4 and discussions relating to these results will be given in section 5.5.

5.2 System modelling using artificial neural networks

BP networks are currently the most popular artificial neural networks and are capable of approximating continuous nonlinear functions to any degree of accuracy. Therefore, in this section BP networks will be used to model three hydraulic systems. The first system demonstrated is a hydraulic transmission system, shown in Figure 5.1 called from now on the 'simulation transmission rig'. This was modelled using a simulation package called Bath*fp* developed in the Fluid Power Centre at the University of Bath[102]. The inputs to the transmission system are the displacements x_p and x_m generated by the servo-systems symbolised as components 3 and 4, which in turn are used to set the fractional displacements of the variable displacement pump and motor. The system outputs are the pump shaft torque, t_p, motor shaft torque, t_m, motor shaft speed, sp_m, the system high pressure between the outlet of the pump and the inlet of the motor, p_h, the system low pressure between the outlet of the motor and the inlet of the pump, p_l, and the flow rate at the outlet of the pump, fl. The input commands, d_p and d_m, to the servo-systems are not shown in the figure and at present the input commands and the displacements from the servo-systems are assumed to be the same. The input and output data sets generated by the software package at different combinations of operation points were collected and used as sets of input and target vectors for training the BP network. After completing the training session, the neural network will be used as a separate model for predicting the outputs of the simulation transmission rig at arbitrarily chosen operation points which the neural network model has never met before. The structure of the BP network used for this purpose is shown in Figure 5.2. The activation function employed in the hidden processing units is the hyperbolic tangent tanh(x) and, for simplicity, not all of the units and connections are drawn in the figure. The data sets for training with their trained results and the prediction data sets with their predicted results are listed in Appendix A. To make the results easily understood, the trained errors for different data sets are shown in Figure 5.3 to Figure 5.8, and the prediction errors are shown in Figure 5.9 to Figure 5.14. The training errors were calculated by comparing the outputs from the trained neural network with the outputs obtained from the simulation transmission rig, using the trained input data sets. The prediction errors were evaluated by comparing the outputs from the trained neural network with the outputs from the simulation transmission rig using input data sets which had never been used during the training session.

Experimental tests were also carried out to check if the same modelling technique works for a real systems. The test rig is shown in Figure 5.15 which will be called the 'test transmission rig'. The structure of the neural network model for the test transmission rig is similar to the one shown in the Figure 5.2, except that the numbers of units in the hidden layer is 111. The experimental data obtained from the test transmission rig subject to displacement commands to the servo-systems were used to train the BP network to mimic the test transmission rig. After the training session was completed, some unused data sets also generated from the test transmission rig were presented to the neural network model to check whether or not it can predict accurate outputs. The training and prediction results are listed in the Appendix B. Again, the training and prediction errors are plotted in Figure 5.16 to Figure 5.27 for each output item. The training errors were obtained by comparing the outputs from the neural network model with the outputs acquired from the test transmission rig and the input data sets to the neural network model were the same as the training ones. The prediction errors were calculated by comparing the outputs from the neural network model with the outputs from the test transmission rig using input data sets which had not been used for training.

The third system to be modelled is a real sequential rig shown in Figure 5.28 called the 'test sequential rig'. The function of this rig is to achieve the outputs shown in the Figure 5.30 to 5.33 and details about the rig can be found in [103]. The input to the neural network model is the sequential time t and the target values corresponding to the input t are the annulus pressure(p_a), piston head pressure(p_p), system pressure(p_s) and displacement(d) of the actuator 1 in the test sequential rig. The structure of the neural network model is shown in Figure 5.29 and the additional input units shown by the dotted-lines will be explained in the last section of this chapter. The results of the neural network model outputs with respect to time t are plotted and shown in Figure 5.30 to Figure 5.33. In each of these figures the outputs from the test sequential rig are plotted using the solid line and the outputs from the neural model are plotted using "+".

5.3 Fault classification of the sequential rig using different artificial neural networks

After establishing the neural network model for the sequential test rig, the model can be used for monitoring the sequential test rig at any time during its process cycle. At a value for time t, the neural network will generate the corresponding outputs at time t+n, for the annulus pressure $\mathbf{p_a}$, the piston head pressure $\mathbf{p_p}$, the system pressure $\mathbf{p_s}$, and the actuator displacement **dis**. These predicted values from the neural network will be assumed to be the correct values of the test sequential rig under normal operational conditions. If any deviation or error, beyond the pre-set limits, of the outputs of the test sequential rig with respect to the normal outputs of the neural network model is detected, then an alarm signal will be triggered by the monitoring system in which the neural network is used as the reference model. This monitoring scheme is shown in Figure 5.34. If any error does occur in the monitored system, then the type of the error must be classified by the monitoring system.

To test the performance of different types of neural networks for fault classification, the faults listed in Table 5.1 were introduced in the test sequential rig. The errors at discrete time intervals are recorded by comparing the outputs of the test sequential rig, in which one type of fault listed in the Table 5.1 is set, with the reference outputs shown in the Figure 5.30 to 5.33. Each set of these errors is called an error pattern vector. There are four sets of error patterns, namely $\mathbf{p}_a$, $\mathbf{p}_p$, $\mathbf{p}_s$, and **dis** error pattern vectors, associated with each fault type or class. A set of error patterns for fault class number 1 are shown in Figure 5.35 to 5.38. In the experiments, each individual error pattern vector has twenty-nine components. Sixty training data sets covering different ranges of faulty settings were utilised for training the BP networks and self-organising maps. These comprised twenty-four sets of real data directly from the rig with the rest created by adding up to 5 percent random noise. It is common to use data sets with added noise signals to train artificial neural networks in order that the trained neural networks can have good performance in a noisy environment. Another thirty sets of untrained data, including twenty-four acquired from the test sequential rig, were used for testing the trained neural networks. The hierarchical structures of the neural networks employed for the test are shown in Figure 5.39. Each of the structures is composed of neural networks arranged in two levels, the upper and lower levels. The difference between the structures (a) and (b) is that all the lower level neural networks in (a) are BP networks and in (b) are SOM networks. The input layer of the BP networks, Figure 5.39(a), in the lower level has twenty-nine units corresponding to the dimension of the error pattern vectors and the number of units in the output layer is set to be ten, equal to the number of fault types.

The two upper level BP networks have ten output units and forty input units. For the hierarchical structure, Figure 5.39 (b), the SOM networks have twenty-nine units in the input layer and a ten by ten output net. Recall that responding to the input pattern the output net of the SOM network produces only one output at one unit. If the SOM network is used for classification, then the output unit represents a specified category to which the input pattern belongs. Should the input pattern be corrupted with noise, then the output unit could be a nearby unit. A hard or crisp partition technique could make a mistake and mis-classify the input pattern. To avoid making crisp decisions, before sending the outputs from the SOM networks to the BP network in the upper level, every output is transformed to ten numbers, similar to fuzzy membership values using equation 5.1. Later the transformed outputs are passed to the BP network in the upper level.

$$
\begin{aligned}
& m = \frac{a - \sqrt{(x - x_{cj})^2 + (y - y_{cj})^2}}{b} \quad \text{if} \;\; 0 \le m \le 1 \\
& m = 0 \quad \text{if} \;\; m < 0 \\
& m = 1 \quad \text{if} \;\; m > 1
\end{aligned} \tag{5.1}
$$

where a and b are positive parameters, x and y are co-ordinates of an output unit on the map, and x_{cj} and y_{cj} are co-ordinates of the centre of the learnt jth fault.

After training the SOM networks, ten sets of functions taking the form of equation (5.1), but with different values of x_{cj} and y_{cj} are generated where each of them corresponds to one of the ten fault classes, which has the class centre located at (x_{cj}, y_{cj}). Thus, for every output of the SOM network, ten membership values are calculated and there are four input patterns at a time. As a result, the BP network in the upper level needs forty input units in it's input layer. Figure 5.40 helps in understanding the membership function associated with equation 5.1. In the figure, the shape of the flat-head cone depends on the values of the parameters a and b and if a point is located between the two dotted-line circles then the membership value of this point, with respect to the fault class with centre at (x_{cj},y_{cj}), is between 0 and 1. The membership value is 0, if the point is on or outside the larger dotted-line circle and is 1, if it is on or inside the small dotted-line circle.

The test outputs using the hierarchical structures shown in the Figure 5.39 are listed in Tables 5.2 and 5.3. Two sets of outputs taken from the Table 5.2 are shown pictorially in Figure 5.41 as examples. In the figure, each of the ten outputs represents an output from the output units and the unit with the highest output symbolises the class to which the input pattern belongs. For example, in Figure 5.41(a) the highest output is from the unit number 5 then we know that the input pattern belongs to the fault class number 5 and in Figure 5.41(b) the highest output is from the unit number 6 then we know that the input pattern belongs the fault class number 6. The detailed outputs from the neural networks in the lower levels in Figure 5.39 are not given here, but the summary of the results are listed in Table 5.4. Table 5.5 shows the summary of the final outcome of classification by using the structures shown in Figure 5.39. In the table two additional test results are included, which were obtained using the structures shown in Figure 5.42(a) and (b). The input patterns for these two alternatives are different from the input patterns for the structures shown in Figure 5.39. The input patterns, for training or for prediction, are acquired by combining the four subsets of data, which are error patterns of p_a, p_p, p_s and d, to one set(combined data) by calculating the square root of $p_{ai}^2+p_{pi}^2+p_{si}^2+d_i^2$, where i is the ith component of each subset. The final outputs from the SOM and BP networks in Figure 5.42 are listed in Table 5.6 and 5.7 respectively. Except for the structures shown in Figure 5.42, the ART2 network was also tested for the same classification task. The classification results obtained using this network are shown in Table 5.8(a),(b) and (c) for different values of the vigilance parameter.

5.4 Fault classification of the simulation transmission rig using fuzzy logic and the artificial neural network

In the second section of this chapter, BP neural networks have been used to predict the outputs of the modelled systems. The trained neural network models can then be used as the healthy state for monitoring the modelled systems. If any output from a modelled system deviates from the neural model output and beyond preset limits, then a fault alarm will be issued from the monitoring system. The error signatures can be used for identifying the fault. The test results of fault classification by making use of the pattern recognition concepts and artificial neural networks were shown in the last section for the test sequential rig. However, for the test transmission rig, real output data for the faulty elements was not available. Therefore, in order to demonstrate the technique of fault classification using fuzzy logic and computational neural-like network, the simulation data produced from the Bath*fp* simulation package, setting faulty components in the simulation transmission rig shown in Figure 5.1, was employed for this purpose. The computational neural-like networks are shown in Figure 5.43 and Figure 5.44, and the complete monitoring system for the simulation rig is shown in Figure 5.45. Figure 5.45 includes the diagnostic subsystem, sometimes called a fuzzy diagnostic subsystem, in which the technique derived from fuzzy logic is practised by the neural-like computational networks. The computational networks in the two figures are different from each other in the third layer which perform fuzzy intersection or algebraic product operation. All of the other units in each corresponding layer of the two structures are identical. The first layer in the structure is the input layer for distributing each component of the input pattern(symptom) to every rule located in the second layer. The rules are shown in the dotted-line boxes. Each of the units in the second layer represent a membership function of a fuzzy set, e.g. small or big, and the outputs from this layer are gated by the units in the third layer, which subsequently produce outputs to the single unit in the last layer by an intersection operation. This is shown in Figure 5.43 by the symbol "$\wedge$", or by an algebraic product operation, shown in Figure 5.44 using the symbol "$\times$". This output unit performs the simplest reasoning by employing the fuzzy union operation and eventually generates the inferential result.

A total of seventeen classes of faults are considered in this trial and associated with these faults there are twenty-nine rules set up for the diagnostic subsystem. The classes of faults are tabulated in Table 5.9 and the rules, which are the heart of the diagnostic subsystem, are listed in Table 5.10. Examples of the rules are as follows:

rule number 0: IF { the error of x_p is small (s) and the error of x_m is small (s) and the error of t_p is small (s) and the error of t_m is small (s) and the error of sp_m is small (s) and the error of p_h is small (s) and the error of p_l is small (s) and the error of fl is small (s) } THEN the fault is the class number 0.

rule number 1: IF { the error of x_p is small (s) and the error of x_m is small (s) and the error of t_p is negatively large (nl) and the error of t_m is negatively large (nl) and the error of sp_m is negatively large (nl) and the error of p_h is negatively large (nl) and the error of p_l is small (s) and the error of fl is negatively large (nl) } THEN the fault is the class number 1-1.

The terms "small (s)", "negatively large (nl)", "positively large (pl)" and "negatively and very large (nvl)" are fuzzy sets for the variable "error" and for each individual fuzzy set a membership function is assigned or generated by adaptive methods[4,5]. The membership functions are shown in Figure 5.46. For the seventeen classes of faults, only six classes were tested by the fuzzy diagnostic subsystem and the total number of test data sets are 201. Since the output data files are large, only samples taken from the outcomes of the tests are listed in Table 5.11, which are tested by applying the fuzzy intersection operation rule on the IF-part of the IF-THEN rules, and Table 5.2, which are tested by applying the algebraic product operation rule to the IF-part of the IF-THEN rules, respectively. The complete test results are listed in Appendix D and E.

The result for the first data set shown in Table 5.11 will be used as an example. The symptom line shows the input pattern vector acquired from the system demonstrated in Figure 5.45 by subtracting the predictive outputs of the sensors from the sensor outputs from the simulation rig. The components of the pattern vector are respectively the errors of pump swash-servo displacement x_p, the motor swash-servo displacement x_m, the torque of the pump shaft t_p, the torque of the motor shaft t_m, the shaft speed of the motor sp_m, the high pressure of the simulation rig p_h, the low pressure of the simulation rig p_l, and the flow rate fl from the pump unit. In the next line the number after "Fault group" is assigned to the testing data for cross-reference and does not influence the results at all. One of the most important data appears after "most suitable rule no.=" and this number gives the final result of the whole inference system, which means that the input pattern satisfies the rule number 11 with the strongest firing strength. The rule 11, referring to Table 5.10, implies that the fault class is the class 5-2. We then need to check Table 5.9 to provide explanation for the fault and the associated faulty part in the monitored rig, and the conclusion is shown in the following line(s) in Table 5.11 and 5.12. IF the user feels that the inference is dubious, the firing strength of all rules are also printed out for reference, which are the outputs from the units in the third layer in Figures 5.43 or 5.44.

5.5 Discussions

Data acquisition was performed using either a 286 or a 386 personal computer with the data acquisition software called SCOPE. Other programs, including the BP networks, SOM networks, ART1(Chapter 3) and ART2, fuzzy min-max neural network(Chapter 4) and the computational neural-like networks, were all written in C codes and compiled by the Turbo C^{++} compiler(Borland), and finally executed on a 486-DX33 personal computer.

Nowadays, commercial software packages are available for some artificial neural networks. However, all programs needed in this research were written by the author for the following reasons.

(a)Programming is the best way to familiarise with different neural networks if this is a new subject to a researcher.

(b)The topic of the artificial neural network has not reached its mature period. A large quantity of papers about training algorithms and the structures of neural networks are published at very fast speed. Commercial software packages usually cover only well developed or popular neural networks, which may or may be not suitable for condition monitoring, and new networks or algorithms can not be found in these packages.

(c)Usually, in an artificial neural network, there are parameters which must be decided by the user and often one needs to investigate the effects of these parameters in different applications. In addition, sometimes new training algorithms for artificial neural networks can be fulfilled by just changing a few codes in old program structures. Commercial packages usually do not provide the environment for these research activities.

(d)In practice, commercial packages can have some problems such as linking with other programs, requiring special hardware and special compilers. If programs are written by the user, these problems can be easily solved. Indeed, a special situation occurred at the beginning of this research. The available computer for this work was a 386-PC and it was shared with other users. Training time for some artificial neural networks was very lengthy and frequently the training session was stopped in the middle of a training course. Commercial packages usually do not include a re-training procedure and after a interruption is made during the training session, the neural network needs to be trained from the very beginning. Therefore, the best solution for this situation was to write programs which can stop and re-start training sessions as one wishes.

The training and prediction errors from the neural network model for the simulation transmission rig, shown in Figures 5.3 to 5.14, are extremely small compared with the errors from the neural network model for the test transmission rig, shown in Figures 5.16 to 5.27. Two causes account for this phenomenon. Firstly, the data sets used for the simulation rig are all larger values compared with the values used in the test rig, which are sensor output voltages. As a result the percentages of the errors are smaller than those for the test rig. Secondly the test rig is not a new one, and worn parts in the rig can cause large output errors. Except for the predicted torque outputs, the majority of predicted outputs are, general speaking, within acceptable error limits. Also, the modelling technique used here ca be compared with the modelling method employed in the original monitoring system[29,30] which was based on an expert system, the former is superior in, at least, two aspects. (i) The current technique can be used to predict the outputs at any suitable operating point. The original monitoring system could only model the transmission system at fixed operating conditions. (ii)For the present technique, the training data was taken from the rig itself, therefore the effects of changing parameters were automatically included in the training data. Using the real data for modelling can guarantee that the trained neural network model indeed represents the real system. As the test transmission rig is pretty old, the component parameters, which must be known in advance for the original monitoring system, are unlikely to be reliable, and consequently it is difficult to acquire very accurate prediction outputs from the monitoring system.

The total number of data sets taken from the test sequential rig for each the four signals p_a, p_p, p_s and d was 1024. Considering that the 1024 sets of data are more than enough for training the BP neural network and some of the data are constant for long periods during the whole cycle, only those regarded as crucial to the shape of the curves are kept as the training data. This makes the sparse points of "+ " in Figures 5.30 to 5.33.

It was very difficult to train the BP network model for the test sequential rig. While the network was being trained the speed of convergence was extremely slow and sometimes it failed to converge. This situation could be caused by the functions which the network were targeted to simulate. It is well known that BP networks are able to approximate continuous nonlinear functions. The difficulty in training may be caused by using only one input, which is the cycle time t, to train four functions in which three are 'step-like' nonlinear functions. After many unsuccessful trials, the neural structure shown in Figure 5.29 was finally chosen and tested. In this neural network structure, two more artificial input units, shown with dotted-lines, were added taking the values of sin(t) and cos(t) as inputs and this structure resulted in the successful outcomes shown in Figure 5.30 to 5.33. There are two reasons for not using feedback signals as part of the inputs to train the neural model. Firstly, for normal operation, the output signals from the sequential test rig were repeatable and a static representation was considered to be suitable for modelling the rig. Secondly, the data sets selected for training the neural network were not acquired at constant time intervals for the reason mentioned in the previous paragraph. As a consequence, feedback inputs were considered not suitable in this case.

Three assumptions were made when the neural networks were used for fault classification for the test sequential rig. The first assumption was that the reference outputs shown in Figure 5.30 to 5.33 are fault free. The second assumption was that only a single fault existed in the test rig, in other words the current effort only tries to classify a single fault detected in the monitored system. The classification of multi-faults was not considered at this time. The third assumption was that the unknown fault patterns were similar to the fault patterns used for training, which are called exemplar patterns. This assumption was made not only because it is inherent in the pattern recognition technique but also because the way the training data was acquired. Using artificial neural networks for classification by the pattern recognition technique needs exemplar patterns for training the neural networks. If the features of an unknown pattern are similar enough to the features of a trained exemplar pattern, then the unknown pattern will be categorised in the same class as the similar exemplar pattern. If the input unknown pattern is not similar to any trained pattern, then mis-classification can occur. Thus, if an artificial neural network is used for the classification purposes, the more exemplar patterns that are used for training the better is the performance of the trained neural network. Also in the experiments, the exemplar data sets and the test data sets, for training and for classification prediction respectively, associated with variant faults were set in limited ranges. For example, if a correct value setting is 4 in a range from 0 to 30, then a faulty setting could be set to be 6 to 10, not too far away from or too close to the correct setting. If the error is too small, corresponding to the faulty setting near the correct setting, the monitoring system could not detect the deviation and the monitored system is regarded as healthy. On the other hand, if the faulty setting deviates too far from the correct setting then the pattern of the error could be very different from the trained exemplar patterns. In this case a new exemplar pattern is needed for training even though the fault actually belongs to the same type of fault which is caused by the same faulty element in the system.

The main goal of this part of investigation was to test the feasibility of applying the artificial neural networks for fault classification in fluid power systems. Therefore, there was no intention to exhaustively cover all possible types of faults that could occur in the monitored system. However, in a practical fault monitoring system it would be necessary to introduce all possible fault patterns for training purposes.

Each of faults caused by the timer malfunctions, fault no.3, no.7 and no.9 in Table 5.1, produced only one error pattern. For fault no.3 and fault no.9, this is taken for granted since if valve 5 is not energised, or permanently energised, there exists only one error pattern respectively. For fault no.7, caused by wrong timer settings, there are many possible patterns, thus the test was carried out on one case only, i.e. at a fixed time of early retract. If it is required, the patterns of wrong timer settings can be taken at suitable time intervals to train the neural network to recognise these fault patterns.

The concept of using combined data sets for the fault classification tests was based on two reasons. Firstly, making use of the hierarchical structures shown in Figure 5.39, the fault classification tasks needs to train six BP networks and four SOM networks. The whole training process is time consuming and after the training is completed the trained weight vectors can require large memory storage. Accordingly, the single network structures, shown in Figure 5.42, were considered as alternatives and for all of the new structures. Secondly, one of the tasks of this study was to compare the performance of the three different types of networks used for fault classification. As the ART networks can accept only positive input patterns, the input patterns must be transformed into another form suitable for all three different networks. The transformation rule was chosen arbitrarily and then tested to see if it meets the requirement for the ART input patterns. The test results summarised in Table 5.5 show that the correct classification rates for the single BP and SOM networks drop significantly compared with the other test results acquired using the hierarchical structures. This is believed to be caused by losses in the features of the data during the data combining process. It is well known that the characteristic features embodied in a pattern is a decisive factor for the success of pattern recognition. The ART2 network is famous for its ability to solve the so-called stability-plasticity dilemma, as mentioned previously in section 3.4. The test results from the ART2 network, listed in Table 5.8(a) ,5.8(b), and 5.8(c), also show that it could categorise the faults into the correct classes if the parameters of the network are correctly chosen. Thus it could be the most suitable network for fault classification. But, there are many parameters in the ART2 network which still need to be understood, and proper values for these parameters are necessary for the successful application of the network.

It is always desirable to reduce the number of sensors employed in a monitoring system. For the sequential system, the tests indicated that it was possible to use only the displacement sensor for the purpose of monitoring, Figure 5.33 shows that the displacement can be predicted accurately and the results in Table 5.4 also show that the rate of correct classification is very high for both networks. These two facts imply that the displacement could contain most of the important features of all fault patterns and could be the best choice to be used for monitoring under the situation where only one sensor is allowed. However, this conjecture can only be proven by further testing.

The computational neural-like networks used in the section 5.4 and shown in Figures 5.43 and 5.44 are often called neural networks in literature[99,104]. As a neural network is considered to have adjustable parameters and training or learning processes are needed, strictly speaking this type of network should not be called a neural network. However, if the parameters which appear in the rules or in the defuzzification processes are determined by learning algorithms, for instance as in the references [86,87], then it is acceptable that the neural-like computational networks can be formally regarded as neural networks.

Since the THEN parts in the IF-THEN rules are singletons, this fact brings about the consequence that the suggested diagnostic subsystem does not need complicated defuzzification techniques and the simple fuzzy union operation can be employed to generate the final inference. This simple union operation allows the computational networks, shown in Figures 5.43 and 5.44, to be simplified. In addition, due to the same reason, no matter which combinations of sup-min or sup-product and intersection or product implication rules are used, the results will be the same. Thus only the operational rules applied for processing the IF parts of the IF-THEN need to be known.

The test results listed in Tables 5.11 and 5.12 and in Appendices D and E show the technique introduced in the last section to be one of the most promising techniques for condition monitoring tasks. The technique can simultaneously make use of both the experts' knowledge and numerical data from the real systems. It does not need a lengthy training course or have the stability-plasticity problem and new information can be added to the suggested system at any time just by expanding it's rule base memories and the number of units in the second layer of the networks as indicated in Figures 5.43 and 5.44. The most notable characteristic of this fuzzy diagnostic subsystem is that once the system is established by the simulation package, it is applicable to the real system immediately. The only thing necessary is to change the shapes of the membership functions, shown in Figure 5.46, or/and to add more rules. The diagnostic mechanism of the original monitoring system[29,30] for the test transmission rig was based on an expert system built upon a commercial software frame. The inferences come from the simple if-then rules and these rules are crisp ones. For the crisp rules, some limits must be set in advance and faults can be detected only if the deviations of the monitored signals pass outside of these limits, otherwise the faulty system is regarded to be healthy. Besides, the if-then rules in the original monitoring system use a one-to-one detecting and diagnostic scheme, previously mentioned in Chapter 1. It checks each sensor one by one along a pre-set searching string(branch). On the other hand, there are no hard limits needed for the fuzzy diagnostic subsystem, and, therefore, it is possible for the fuzzy diagnostic subsystem to detect incipient faults. Also, for the fuzzy diagnostic subsystem the inference conclusions are based on considering the deviations of all signals simultaneously. Thus the diagnostic scheme of the fuzzy diagnostic subsystem is not of the one-to-one type and, as a result it could make use of less sensors to diagnose more faults than the existing monitoring system. Another important feature of the fuzzy diagnostic subsystem is the sensitivity of the diagnosis can be easily adjusted by modifying the shape of the membership functions to achieve a compromise between mis-classification and detecting incipient faults.

In this research, there were seventeen artificial neural networks used in total, including seven BP networks, five SOM networks, one ART1, one ART2, one fuzzy network and two neural-like computational networks. To train the BP and SOM neural networks was time consuming, not because the training time needed for high accuracy was lengthy but also because there was no universal method of determining the structure of a neural network and trial-and-error methods were frequently employed. For example, the neural network used for simulating the test sequential rig, shown in Figure 5.29, was tested for many times in a period of three months before the final structure of this network was set and the mean squared error of the outputs was sufficiently small. The final results of this network are shown in Figures 5.30 to 5.33. For this network, the total training cycles, using a 486DX33 personal computer and 118 sets of training data, was 22,140 and the equivalent training time was nearly 23 hours. As mentioned previously, failures frequently occurred during BP network training session and in this research three types of failures were encountered. These failures can be summarised into three categories, which are slow convergence rate, fluctuations in output error and failing to converge. Figure 5.47 shows the types of failures encountered during training BP networks. When any of these failures does happen, the training rate or the structure of the network must be changed by trial-and-error techniques.

The technique proposed for the SOM, shown in Figures 5.39(b), 5.40 and 5.42, may not be suitable for some forms of clusters. However, it performed well in the tests for the sequential test rig. For comparison, tests using the original hard decision technique employed by the SOM were also done. To use the original technique, the BP network shown in Figure 5.39(b) was replaced by a voting decision making scheme and the prediction results, shown in Appendix F (Figures e to h and Figure j), were as good as the proposed technique except in one case(Figure g). In this, comparing with the last row in Table 5.4, there was no misclassification, but the final conclusion results were the same as those shown in the last column of Tables 5.3 and 5.6.

Table 5.1 Definitions of faulty settings.

	faulty settings	symptoms
fault no.1	valve 16 set too low	p_a too low and forward speed too fast
fault no.2	valve 10 open too much	second part of extend too fast
fault no.3	valve 5 not energised	extend continuous at fast speed
fault no.4	restrictor 2A open too much	decompression too fast
fault no.5	valve 23 set too high	initial retract too fast
fault no.6	interference between valves 22 and 23	p_a,p_n and p_s changing abruptly and unsteady displacement
fault no.7	timer 16 set too short	cycle not completed
fault no.8	valve 20 set too low	p_s too low and extend too slow
fault no.9	valve 5 permanently energised	no initial fast approach
fault no.10	valve 17 set too high	extend too slow

Table 5.2 Outputs from the BP-BP upper level network shown in Figure 5.39(a)

class no.	output from the unit no.									
	1	2	3	4	5	6	7	8	9	10	
1	0.88	0.09	0.11	0.10	0.10	0.09	0.10	0.10	0.10	0.11	s
1	0.89	0.11	0.10	0.09	0.09	0.09	0.10	0.10	0.10	0.11	s
1	0.89	0.12	0.09	0.09	0.09	0.09	0.10	0.10	0.10	0.11	s
2	0.16	0.87	0.10	0.07	0.12	0.10	0.10	0.09	0.09	0.09	s
2	0.16	0.84	0.13	0.16	0.08	0.09	0.09	0.12	0.10	0.09	s
2	0.07	0.89	0.08	0.10	0.15	0.10	0.11	0.09	0.10	0.12	s
3	0.10	0.10	0.89	0.11	0.10	0.10	0.10	0.09	0.10	0.10	s
3	0.10	0.10	0.90	0.10	0.11	0.10	0.10	0.08	0.10	0.10	s
3	0.10	0.10	0.90	0.10	0.10	0.10	0.10	0.08	0.10	0.10	s
4	0.13	0.10	0.10	0.89	0.10	0.10	0.10	0.10	0.10	0.10	s
4	0.08	0.11	0.14	0.89	0.09	0.10	0.09	0.11	0.09	0.10	s
4	0.17	0.08	0.09	0.88	0.11	0.10	0.12	0.09	0.10	0.10	s
5	0.10	0.09	0.09	0.09	0.90	0.09	0.11	0.10	0.10	0.11	s
5	0.07	0.13	0.09	0.17	0.77	0.12	0.11	0.10	0.10	0.11	s
5	0.11	0.10	0.11	0.09	0.88	0.11	0.09	0.10	0.10	0.10	s
6	0.08	0.14	0.12	0.10	0.15	0.86	0.10	0.09	0.10	0.11	s
6	0.08	0.12	0.11	0.12	0.11	0.88	0.10	0.10	0.11	0.09	s
6	0.09	0.19	0.14	0.12	0.09	0.80	0.12	0.10	0.14	0.11	s
7	0.10	0.11	0.10	0.11	0.10	0.10	0.89	0.11	0.11	0.10	s
7	0.10	0.10	0.11	0.10	0.11	0.10	0.89	0.10	0.10	0.10	s
7	0.11	0.09	0.12	0.09	0.12	0.10	0.89	0.09	0.10	0.10	s
8	0.10	0.12	0.07	0.11	0.10	0.11	0.10	0.89	0.11	0.09	s
8	0.12	0.10	0.08	0.10	0.10	0.10	0.10	0.89	0.10	0.10	s
8	0.09	0.08	0.10	0.10	0.10	0.10	0.11	0.88	0.10	0.13	s
9	0.09	0.11	0.10	0.11	0.10	0.10	0.10	0.10	0.90	0.10	s
9	0.09	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.90	0.11	s
9	0.09	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.90	0.11	s
10	0.12	0.09	0.09	0.11	0.09	0.14	0.10	0.09	0.10	0.90	s
10	0.10	0.09	0.09	0.11	0.10	0.11	0.11	0.10	0.10	0.90	s
10	0.10	0.07	0.12	0.10	0.15	0.08	0.10	0.11	0.11	0.87	s

s=successful classification u=unsuccessful classification

Table 5.3 Outputs from the SOM-BP upper level network shown in Figure 5.39(b)

class no.	output from the unit no.									
	1	2	3	4	5	6	7	8	9	10	
1	0.88	0.11	0.08	0.13	0.09	0.08	0.10	0.11	0.10	0.09	s
2	0.12	0.90	0.10	0.09	0.10	0.13	0.08	0.10	0.10	0.11	s
3	0.15	0.09	0.86	0.14	0.09	0.08	0.09	0.15	0.07	0.12	s
4	0.10	0.09	0.10	0.89	0.10	0.09	0.10	0.10	0.10	0.10	s
5	0.10	0.10	0.10	0.10	0.90	0.10	0.10	0.10	0.10	0.10	s
6	0.09	0.09	0.10	0.10	0.13	0.88	0.10	0.10	0.13	0.09	s
7	0.10	0.10	0.10	0.10	0.11	0.13	0.89	0.09	0.09	0.10	s
8	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.90	0.10	0.10	s
9	0.11	0.10	0.10	0.10	0.10	0.10	0.14	0.09	0.90	0.10	s
10	0.09	0.10	0.14	0.09	0.10	0.10	0.10	0.10	0.12	0.89	s
1	0.89	0.10	0.10	0.09	0.10	0.08	0.10	0.10	0.10	0.09	s
1	0.87	0.13	0.08	0.11	0.09	0.08	0.10	0.11	0.10	0.10	s
2	0.74	0.32	0.15	0.09	0.10	0.08	0.08	0.10	0.10	0.08	u
2	0.12	0.88	0.10	0.13	0.11	0.17	0.07	0.09	0.11	0.08	s
3	0.10	0.10	0.90	0.10	0.10	0.11	0.10	0.10	0.10	0.10	s
3	0.12	0.11	0.88	0.10	0.09	0.09	0.09	0.09	0.09	0.14	s
4	0.10	0.08	0.09	0.89	0.09	0.10	0.09	0.10	0.15	0.09	s
4	0.08	0.11	0.10	0.87	0.11	0.09	0.11	0.10	0.10	0.12	s
5	0.12	0.13	0.11	0.06	0.89	0.10	0.10	0.11	0.11	0.12	s
5	0.13	0.13	0.10	0.08	0.89	0.11	0.12	0.10	0.10	0.08	s
6	0.09	0.09	0.11	0.10	0.10	0.89	0.10	0.14	0.10	0.09	s
6	0.06	0.15	0.12	0.05	0.32	0.67	0.13	0.10	0.13	0.13	s
7	0.10	0.12	0.10	0.10	0.10	0.10	0.90	0.10	0.10	0.10	s
7	0.09	0.13	0.11	0.11	0.10	0.09	0.88	0.11	0.09	0.10	s
8	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.90	0.10	0.09	s
8	0.09	0.09	0.09	0.10	0.10	0.10	0.14	0.88	0.10	0.09	s
9	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.90	0.10	s
9	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.90	0.10	s
10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.90	s
10	0.12	0.08	0.09	0.12	0.05	0.11	0.11	0.11	0.15	0.89	s

s=successful classification, u=unsuccessful classification

Table 5.4 Summary of classification results from lower level networks

network structure	sample no.	p_a pattern		p_n pattern		p_s pattern		displ. pattern	
		s	u	s	u	s	u	s	u
BP-BP,Fig.5.39(a)	30	30	0	28	2	29	1	30	0
SOM-BP,Fig.5.39(b)	30	29	1	29	1	29	1	29	1

s=successful classification u=unsuccessful classification

Table 5.5 Summary of the final classification results

network structure	no. of samples	data type	no. of correct classification
BP-BP, Fig.5.39(a)	30	original	30 (100%)
SOM-BP, Fig.5.39(b)	30	original	30 (96.7%)
SOM, Fig.5.42(a)	30	combined	26 (86.7%)
BP, Fig.5.42(b)	30	combined	24 (80%)

Table 5.6 Outputs from single level SOM network shown in Figure 5.42(a)

sample no.	outputs									
	1	2	3	4	5	6	7	8	9	10	
1	0.59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	s
2	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	s
3	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	s
4	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	s
5	0.67	0.13	0.00	0.13	0.00	0.00	0.00	0.00	0.00	0.00	u
6	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	s
7	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.59	0.00	s
8	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.59	0.00	s
9	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.59	0.00	s
10	0.67	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.33	0.00	s
11	0.33	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.28	0.00	s
12	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.13	0.00	s
13	0.00	0.13	0.00	0.13	0.67	0.13	0.28	0.00	0.00	0.00	s
14	0.00	0.28	0.00	0.00	1.00	0.28	0.13	0.00	0.00	0.00	s
15	0.00	0.28	0.00	0.00	1.00	0.28	0.13	0.00	0.00	0.00	s
16	0.00	0.00	0.00	0.00	0.13	0.67	1.00	0.00	0.00	0.00	u
17	0.00	0.00	0.00	0.00	0.13	0.67	1.00	0.00	0.00	0.00	u
18	0.00	0.00	0.00	0.00	0.13	0.67	1.00	0.00	0.00	0.00	u
19	0.00	0.00	0.00	0.00	0.00	0.33	1.00	0.33	0.00	0.00	s
20	0.00	0.00	0.00	0.00	0.00	0.33	1.00	0.33	0.00	0.00	s
21	0.00	0.00	0.00	0.00	0.00	0.33	1.00	0.33	0.00	0.00	s
22	0.00	0.00	0.00	0.00	0.00	0.00	0.33	1.00	0.00	0.33	s
23	0.00	0.00	0.00	0.00	0.00	0.00	0.33	1.00	0.00	0.33	s
24	0.00	0.00	0.00	0.00	0.00	0.00	0.28	1.00	0.00	0.28	s
25	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	1.00	0.00	s
26	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	1.00	0.00	s
27	0.00	0.00	0.59	0.00	0.00	0.00	0.00	0.00	1.00	0.00	s
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28	0.00	1.00	s
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28	0.00	1.00	s
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.67	s

s=successful classification u=unsuccessful classification

Table 5.7 Outputs from single level BP network shown in Figure 5.42(b)

class no.	output from the unit no.									
	1	2	3	4	5	6	7	8	9	10	
1	0.85	0.09	0.15	0.10	0.10	0.11	0.09	0.10	0.11	0.11	s
1	0.89	0.12	0.10	0.10	0.09	0.10	0.10	0.10	0.10	0.10	s
1	0.93	0.11	0.07	0.11	0.09	0.10	0.11	0.09	0.09	0.10	s
2	0.18	0.91	0.05	0.05	0.11	0.13	0.10	0.10	0.10	0.13	s
2	0.24	0.72	0.18	0.25	0.04	0.07	0.11	0.13	0.06	0.07	s
2	0.02	0.96	0.06	0.10	0.19	0.13	0.10	0.07	0.13	0.14	s
3	0.07	0.04	0.82	0.20	0.10	0.10	0.14	0.36	0.15	0.05	s
3	0.07	0.04	0.80	0.19	0.10	0.11	0.14	0.34	0.16	0.05	s
3	0.08	0.04	0.77	0.18	0.11	0.12	0.13	0.34	0.19	0.05	s
4	0.15	0.09	0.07	0.87	0.10	0.11	0.10	0.10	0.12	0.10	s
4	0.04	0.11	0.18	0.93	0.08	0.09	0.09	0.11	0.11	0.09	s
4	0.19	0.08	0.07	0.88	0.12	0.11	0.11	0.09	0.08	0.11	s
5	0.07	0.08	0.15	0.13	0.93	0.05	0.13	0.09	0.10	0.09	s
5	0.07	0.15	0.07	0.15	0.83	0.17	0.08	0.11	0.09	0.12	s
5	0.19	0.09	0.06	0.05	0.90	0.15	0.09	0.10	0.15	0.12	s
6	0.05	0.03	0.80	0.33	0.04	0.17	0.65	0.07	0.06	0.05	u
6	0.05	0.04	0.82	0.32	0.03	0.18	0.67	0.06	0.05	0.04	u
6	0.05	0.04	0.81	0.27	0.03	0.20	0.67	0.06	0.06	0.04	u
7	0.06	0.06	0.49	0.09	0.10	0.16	0.60	0.06	0.17	0.06	s*
7	0.06	0.06	0.53	0.09	0.10	0.17	0.61	0.06	0.17	0.06	s*
7	0.06	0.05	0.56	0.09	0.10	0.16	0.63	0.06	0.17	0.05	s*
8	0.09	0.13	0.04	0.14	0.09	0.09	0.08	0.95	0.08	0.11	s
8	0.11	0.12	0.05	0.10	0.12	0.09	0.06	0.95	0.08	0.12	s
8	0.08	0.09	0.31	0.09	0.08	0.12	0.19	0.65	0.14	0.09	s
9	0.04	0.04	0.66	0.05	0.16	0.15	0.31	0.08	0.53	0.06	u
9	0.05	0.04	0.67	0.05	0.17	0.16	0.30	0.08	0.52	0.06	u
9	0.04	0.05	0.70	0.05	0.16	0.16	0.31	0.08	0.50	0.05	u
10	0.07	0.20	0.02	0.26	0.06	0.12	0.10	0.07	0.06	0.98	s
10	0.07	0.14	0.11	0.13	0.07	0.11	0.13	0.10	0.08	0.89	s
10	0.14	0.05	0.28	0.04	0.16	0.09	0.09	0.11	0.19	0.70	s

s*: ambiguous results s=successful classification u=unsuccessful classification

Table 5.8(a) Categories of faults by ART2 (ρ=0.98)

sample no.	faulty setting	category no.
1,2,3	no.1	0,0,0
4,6	no.2	1,1
5	no.2	0
7,8,9	no.3	2,2,2
10,11,12	no.4	3,3,3
13,14,15	no.5	4,4,4
16,17,18	no.6	5,5,5
19,20,21	no.7	5,5,5
22,23	no.8	6,6
24	no.8	7
25,26,27	no.9	8,8,8
28,29,30	no.10	9,9,9

Table 5.8(b) Categories of faults by ART2 (ρ=0.99)

sample no.	faulty setting	category no.
1,2,3	no.1	0,0,0
4	no.2	1
5	no.2	0
6	no.2	2
7,8,9	no.3	3,3,3
10,11,12	no.4	4,4,4
13	no.5	5
14,15	no.5	6,6
16,17,18	no.6	7,7,7
19,20,21	no.7	8,8,8
22,23	no.8	9,9
24	no.8	10
25,26,27	no.9	11,11,11
28	no.10	12
29	no.10	13
30	no.10	14

Table 5.8(c) Categories of faults by ART2 (ρ=0.995)

Sample no.	faulty setting	category no.
1,2,3	no.1	0,0,0
4	no.2	1
5	no.2	2
6	no.2	3
7,8,9	no.3	4,4,4
10	no.4	5
11,12	no.4	6,6
13	no.5	7
14,15	no.5	8,8
16,17,18	no.6	9,9,9
19,20,21	no.7	10,10,10
22,23	no.8	11,11
24	no.8	12
25,26,27	no.9	13,13,13
28	no.10	14
29	no.10	15
30	no.10	16

Table 5.9 The definitions and descriptions of fault classes.

class no.	**type of fault**	**description**
0	no fault	operating under normal condition
1-1,1-2,1-3	over-leak in the pump unit	too much internal leakage
2-1,2-2	over-leak in the motor unit	too much internal leakage
2-1	leak in the pipe	leak in the pipe between the outlet of the pump and the inlet of the motor
3-1,3-2	over-loaded on the motor shaft	motor shaft over-loaded
4-1,4-2	fault in the pump swash-servo system	servo-system failing to follow the swash-plate input commands
5-1,5-2	fault in the motor swash-servo system	servo-system failing to follow the swash-plate input commands
6-1,6-2	fault in the pump servo sensor	swash-plate servo-system displacement sensor being faulty
7-1,7-2	fault in the motor servo sensor	swash-plate servo-system displacement sensor being faulty
8-1,8-2	fault in the pump torque sensor	pump torque sensor being faulty
9-1,9-2	fault in the motor torque sensor	motor torque sensor being faulty
2-2,10	fault in the motor speed sensor	motor speed sensor being faulty
11-1,11-2	fault in the high pressure sensor	high pressure sensor being faulty
12-1,12-2	fault in the low pressure sensor	low pressure sensor being faulty
1-3,13	fault in the flow sensor	flow sensor being faulty
2-2	high pressure setting too low	high pressure setting too low
0,14	high pressure setting too high	high pressure setting too high
15	low pressure setting too high	low pressure setting too high
16	low pressure setting too low	low pressure setting too low
17	system power failure	system power failure or not switched on

Table 5.10 The IF-THEN rules for the fuzzy diagnostic system.

rule no.	IF								THEN
	x_n	x_m	t_n	t_m	sp_m	p_h	p_l	fl	
0	s	s	s	s	s	s	s	s	It is the fault class 0.
1	s	s	nl	nl	nl	nl	s	nl	It is the fault class 1-1.
2	s	s	s	s	nl	s	s	nl	It is the fault class 1-2.
3	s	s	s	s	s	s	s	nl	It is the fault class 1-3.
4	s	s	nl	nl	nl	nl	s	s	It is the fault class 2-1.
5	s	s	s	s	nl	s	s	s	It is the fault class 2-2.
6	s	s	pl	pl	s	pl	s	s	It is the fault class 3-1.
7	s	s	s	pl	nl	s	s	s	It is the fault class 3-2.
8	nl	s	nl	nl	nl	nl	s	nl	It is the fault class 4-1.
9	pl	s	pl	pl	pl	pl	s	pl	It is the fault class 4-2.
10	s	nl	pl	pl	pl	pl	s	s	It is the fault class 5-1.
11	s	pl	nl	nl	nl	nl	s	s	It is the fault class 5-2.
12	pl	s	s	s	s	s	s	s	It is the fault class 6-1.
13	nl	s	s	s	s	s	s	s	It is the fault class 6-2.
14	s	pl	s	s	s	s	s	s	It is the fault class 7-1.
15	s	nl	s	s	s	s	s	s	It is the fault class 7-2.
16	s	s	pl	s	s	s	s	s	It is the fault class 8-1.
17	s	s	nl	s	s	s	s	s	It is the fault class 8-2.
18	s	s	s	pl	s	s	s	s	It is the fault class 9-1.
19	s	s	s	nl	s	s	s	s	It is the fault class 9-2.
20	s	s	s	s	pl	s	s	s	It is the fault class 10.
21	s	s	s	s	s	pl	s	s	It is the fault class 11-1.
22	s	s	s	s	s	nl	s	s	It is the fault class 11-2.
23	s	s	s	s	s	s	pl	s	It is the fault class 12-1.
24	s	s	s	s	s	s	nl	s	It is the fault class 12-2.
25	s	s	s	s	s	s	s	pl	It is the fault class 13.
26	s	s	nvl	nvl	nvl	nvl	nvl	nvl	It is the fault class 17.
27	s	s	nl	nl	nl	s	pl	s	It is the fault class 15.
28	s	s	pl	pl	pl	s	nl	s	It is the fault class 16.
29	s	s	pl	pl	pl	pl	s	s	It is the fault class 14.

Table 5.11 Some samples of the final outputs of the diagnostic system by applying the fuzzy intersection operation to the IF-part of the IF-THEN rules in the fuzzy inferring processes.

```
***** data no:1 *****
Symptoms:  0.000  2.000  -1.963  -1.507  -1.884  -2.470  0.006  0.027
Faulty group=6  Most suitable rule no.=11
The most possible faulty part in the system is the motor swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.177 0.000 0.000 0.000 0.333 0.177 0.000 0.000 0.000 0.000
 0.000 0.502 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.002 0.000 0.000
***** data no:2 *****
Symptoms:  0.000 -2.000  1:665  1.646  2.127  2.573  0.210 -0.023
Faulty group=6  Most suitable rule no.=10
The most possible faulty part in the system is the motor swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.142 0.000 0.000 0.008 0.000 0.000 0.291 0.000 0.000 0.000
 0.549 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:3 *****
Symptoms:  2.000  0.000  3.150  1.622  1.947  1.594  0.202  2.115
Faulty group=5  Most suitable rule no.=9
The most possible faulty part in the system is the pump swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.000 0.000 0.000 0.000 0.000 0.000 0.295 0.000 0.000 0.531
 0.000 0.000 0.000 0.000 0.000 0.000 0.021 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.295
***** data no:4 *****
Symptoms: -2.000  0.000 -3.054 -1.403 -1.742 -1.415 -0.066 -2.029
Faulty group=5  Most suitable rule no.=8
The most possible faulty part in the system is the pump swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.000 0.333 0.000 0.000 0.324 0.000 0.000 0.000 0.468 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.008 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:5 *****
Symptoms:  0.000  0.000 -0.010  0.182  0.182  0.159  0.188 -0.003
Faulty group=0  Most suitable rule no.=0
The monitored system is operating in normal condition.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.937 0.000 0.000 0.001 0.000 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:6 *****
Symptoms:  0.000  0.000  0.057  0.190 -0.023 -0.055  0.200 -0.050
Faulty group=0  Most suitable rule no.=0
The monitored system is operating in normal condition.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.933 0.000 0.008 0.017 0.000 0.008 0.000 0.008 0.000 0.000
```

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

***** data no:7 *****
Symptoms: 0.000 0.000 -1.202 -1.265 -1.548 -1.461 0.123 -3.036
Faulty group=1 Most suitable rule no.=2
The most possible fault in the system is over-leak in the pump unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.401 0.513 0.484 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:8 *****
Symptoms: 0.000 0.000 -0.948 -1.098 -1.406 -1.129 0.082 -1.653
Faulty group=1 Most suitable rule no.=3
The most possible fault in the system is over-leak in the pump unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.449 0.316 0.469 0.531 0.316 0.449 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.027 0.000 0.000
***** data no:9 *****
Symptoms: 0.000 0.000 -2.217 -2.541 -3.181 -2.722 -0.147 -3.271
Faulty group=1 Most suitable rule no.=1
The most possible fault in the system is over-leak in the pump unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.739 0.093 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:10 *****
Symptoms: 0.000 0.000 -1.154 -1.337 -1.709 -1.381 0.039 0.002
Faulty group=2 Most suitable rule no.=5
The most possible fault in the system is over-leak in the motor unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.430 0.000 0.000 0.000 0.385 0.540 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.013 0.000 0.000
***** data no:11 *****
Symptoms: 0.000 0.000 -1.555 -1.684 -2.017 -1.883 0.123 0.190
Faulty group=2 Most suitable rule no.=4
The most possible fault in the system is over-leak in the motor unit or in the circuit before the motor unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.328 0.000 0.000 0.000 0.518 0.372 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.041 0.000 0.000
***** data no:12 *****
Symptoms: 0.000 0.000 -6.494 -7.454 -9.118 -7.982 -0.805 0.034
Faulty group=3 Most suitable rule no.=4
The most possible fault in the system is over-leak in the motor unit or in the circuit before the motor unit.

If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.000 0.000 0.000 0.732 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
(continued in the next page)
***** data no:13 *****
Symptoms: 0.000 0.000 -3.586 -5.073 -7.268 -5.075 -0.538 -0.011
Faulty group=3 Most suitable rule no.=4
The most possible fault in the system is over-leak in the motor unit or in the circuit before the motor unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.004 0.000 0.000 0.821 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:14 *****
Symptoms: 0.000 0.000 0.541 10.730 -6.475 0.584 -0.537 0.163
Faulty group=4 Most suitable rule no.=7
The most possible fault in the system is over-loaded.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.805 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:15 *****
Symptoms: 0.000 0.000 5.767 17.463 -0.128 7.030 -0.147 -0.131
Faulty group=4 Most suitable rule no.=6
The most possible fault in the system is over-loaded.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.000 0.000 0.000 0.000 0.000 0.951 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:16 *****
Symptoms: 0.000 0.000 -99.000 -50.000 -50.000 -50.000 -50.000 -99.000
Faulty group=7 Most suitable rule no.=26
The most possible fault in the system is the system power.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000

Table 5.12 Some samples of the final outputs of the diagnostic system by applying the algebraic product operation rule to the IF-part of the IF-THEN rules in the fuzzy inferring processes.

```
***** data no:1 *****
Symptom: 0.000000 2.000000 -1.962820 -1.506530 -1.883810 -2.469810 0.005670 0.027370
Faulty group=6  Most suitable rule no.=11
The most possible faulty part in the system is the motor swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.004  0.000  0.000  0.000  0.056  0.006  0.000  0.000  0.000  0.000
 0.000  0.112  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
***** data no:2 *****
Symptom: 0.000000 -2.000000 1.664640 1.645840 2.127480 2.572970 0.210240 -0.022980
Faulty group=6  Most suitable rule no.=10
The most possible faulty part in the system is the motor swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.003  0.000  0.000  0.000  0.000  0.000  0.023  0.000  0.000  0.000
 0.114  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.057
***** data no:3 *****
Symptom: 2.000000 0.000000 3.149510 1.621960 1.946690 1.593990 0.202480 2.115000
Faulty group=5  Most suitable rule no.=9
The most possible faulty part in the system is the pump swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.000  0.000  0.000  0.000  0.000  0.000  0.009  0.000  0.000  0.082
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.017
***** data no:4 *****
Symptom: -2.000000 0.000000 -3.054220 -1.403290 -1.741580 -1.415020 -0.065610 -2.029490
Faulty group=5  Most suitable rule no.=8
The most possible faulty part in the system is the pump swash-servo system.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.000  0.028  0.000  0.000  0.014  0.000  0.000  0.000  0.057  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
***** data no:5 *****
Symptom: 0.000000 0.000000 -0.009560 0.181930 0.182040 0.159170 0.188010 -0.002510
Faulty group=0  Most suitable rule no.=0
The monitored system is operating in normal condition.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.780  0.000  0.000  0.001  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
***** data no:6 *****
Symptom: 0.000000 0.000000 0.056530 0.190360 -0.022800 -0.055280 0.199750 -0.049820
Faulty group=0  Most suitable rule no.=0
The monitored system is operating in normal condition.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the
following firing strength of each rule.
 0.822  0.000  0.000  0.014  0.000  0.006  0.000  0.000  0.000  0.000
```

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:7 *****
Symptom: 0.000000 0.000000 -1.202240 -1.264910 -1.547630 -1.461140 0.123350 -3.035560
Faulty group=1 Most suitable rule no.=2
The most possible fault in the system is over-leak in the pump unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.041 0.088 0.083 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:8 *****
Symptom: 0.000000 0.000000 -0.948050 -1.097540 -1.406040 -1.128670 0.082230 -1.653270
Faulty group=1 Most suitable rule no.=3
The most possible fault in the system is over-leak in the pump unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.063 0.011 0.068 0.077 0.009 0.055 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:9 *****
Symptom: 0.000000 0.000000 -2.216920 -2.540910 -3.181400 -2.722430 -0.147040 -3.270940
Faulty group=1 Most suitable rule no.=1
The most possible fault in the system is over-leak in the pump unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.540 0.004 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:10 *****
Symptom: 0.000000 0.000000 -1.154440 -1.337110 -1.709380 -1.381140 0.038800 0.001600
Faulty group=2 Most suitable rule no.=5
The most possible fault in the system is over-leak in the motor unit
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.078 0.000 0.000 0.000 0.044 0.103 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:11 *****
Symptom: 0.000000 0.000000 -1.555480 -1.684100 -2.016540 -1.883410 0.123350 0.189900
Faulty group=2 Most suitable rule no.=4
The most possible fault in the system is over-leak in the motor unit or in the circuit before the motor unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.023 0.000 0.000 0.000 0.110 0.047 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.003 0.000 0.000
***** data no:12 *****
Symptom: 0.000000 0.000000 -6.494030 -7.454220 -9.118270 -7.982320 -0.805220 0.034500
Faulty group=3 Most suitable rule no.=4
The most possible fault in the system is over-leak in the motor unit or in the circuit before the motor unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the

following firing strength of each rule.
0.000 0.000 0.000 0.000 0.723 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:13 *****
Symptom: 0.000000 0.000000 -3.585900 -5.072560 -7.268250 -5.074540 -0.538190 -0.010680
Faulty group=3 Most suitable rule no.=4
The most possible fault in the system is over-leak in the motor unit or in the circuit before the motor unit.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.003 0.000 0.000 0.818 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:14 *****
Symptom: 0.000000 0.000000 0.541150 10.730340 -6.475050 0.583530 -0.536610 0.163130
Faulty group=4 Most suitable rule no.=7
The most possible fault in the system is over-loaded.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.513 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:15 *****
Symptom: 0.000000 0.000000 5.767470 17.463289 -0.127920 7.030280 -0.147040 -0.130780
Faulty group=4 Most suitable rule no.=6
The most possible fault in the system is over-loaded.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.000 0.000 0.000 0.000 0.000 0.871 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:16 *****
Symptom: 0.000000 0.000000 -99.000000 -50.000000 -50.000000 -50.000000 -50.000000 -99.000000
Faulty group=7 Most suitable rule no.=26
The most possible fault in the system is the system power.
If the diagnostic conclusion is dubious, other possible fault(s) could be found by checking the following firing strength of each rule.
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000

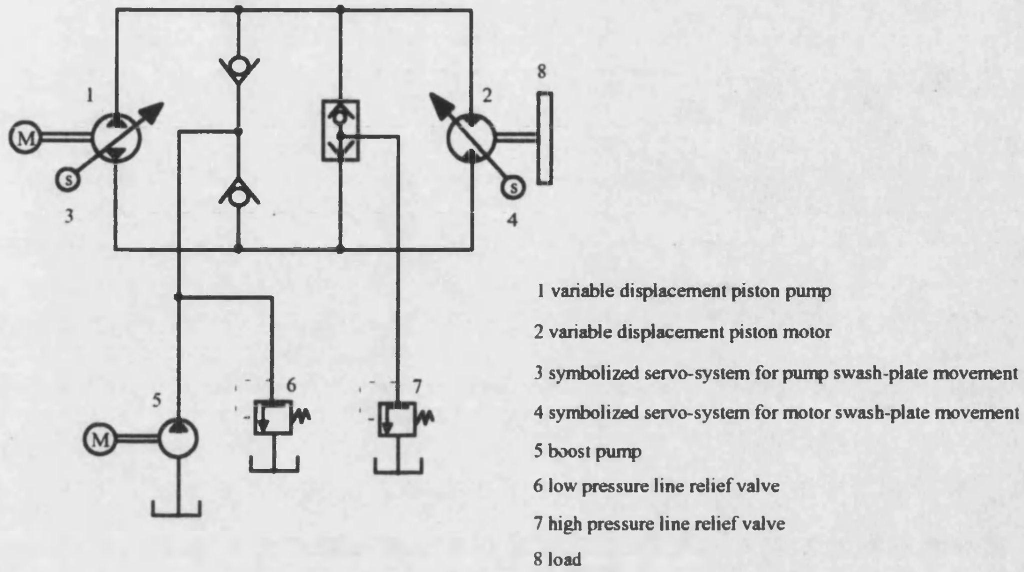


Figure 5.1 Diagram of the simulation transmission rig.

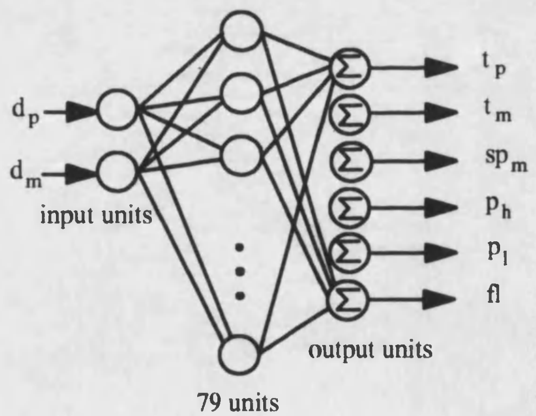


Figure 5.2 The structure of the neural model for the simulation transmission rig.

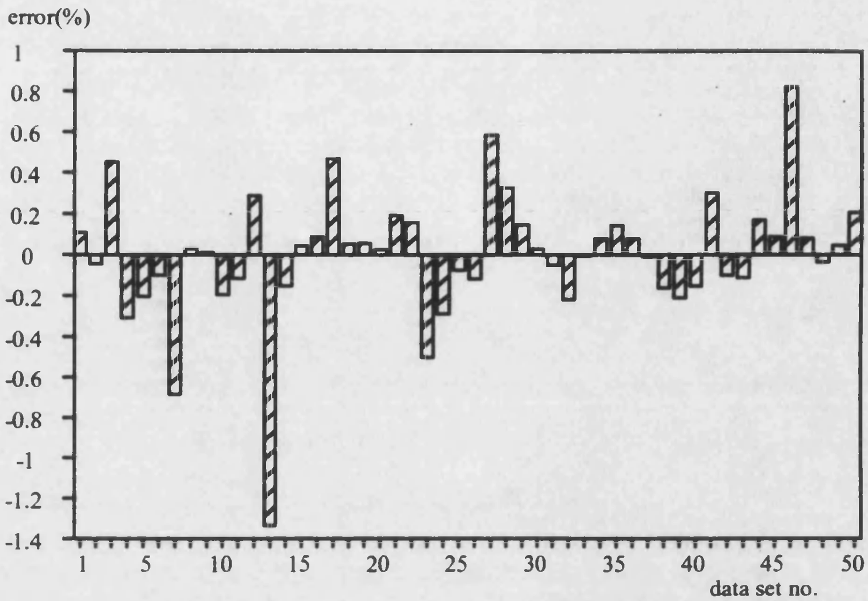


Figure 5.3 Pump torque training errors for the simulation transmissin rig.

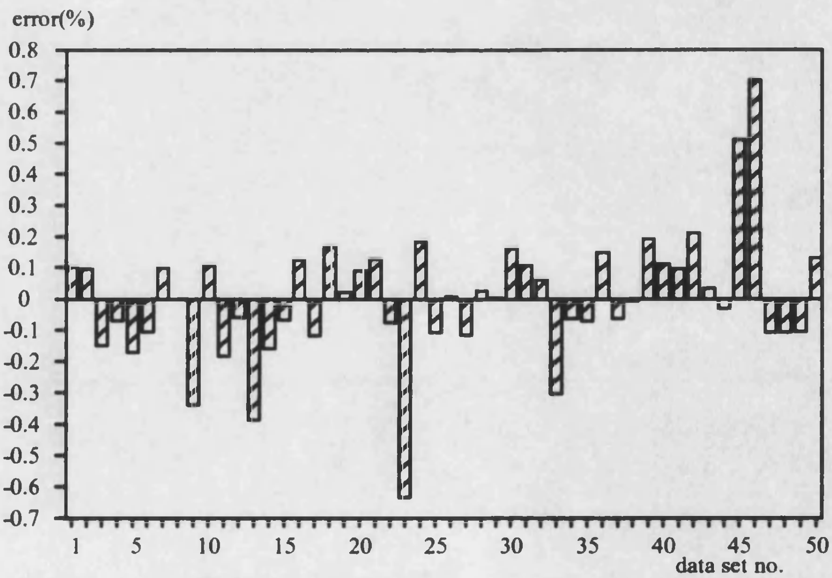


Figure 5.4 Motor torque training errors for the simulation transmission rig.

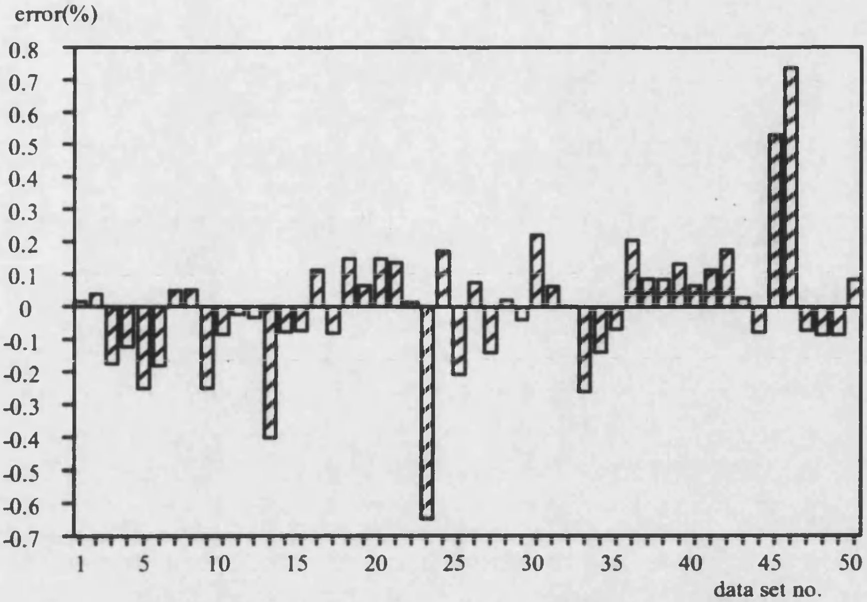


Figure 5.5 Motor speed training errors for the simulation transmission rig.

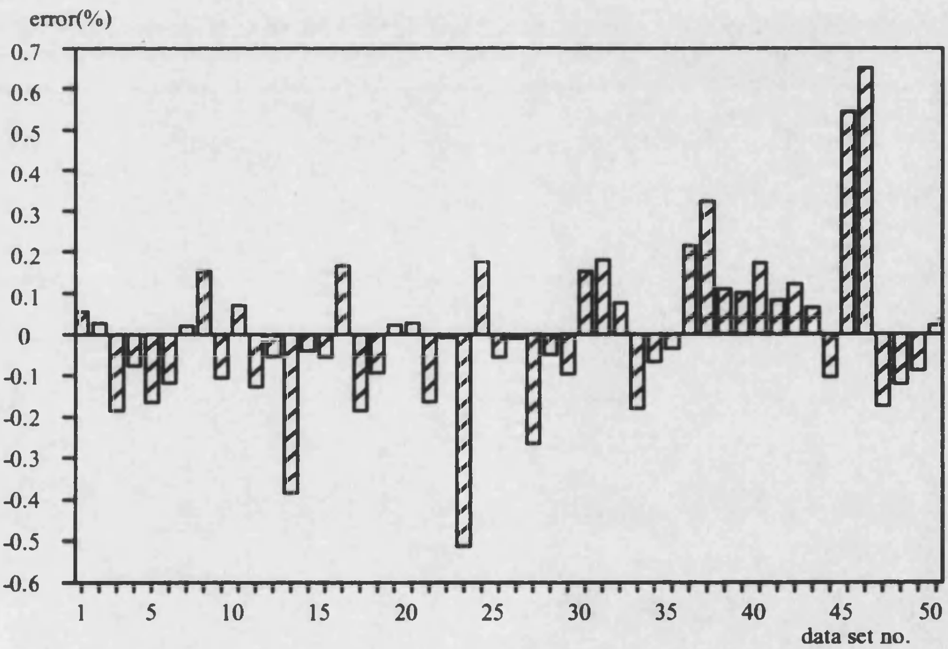


Figure 5.6 High pressure training errors for the simulation transmission rig.

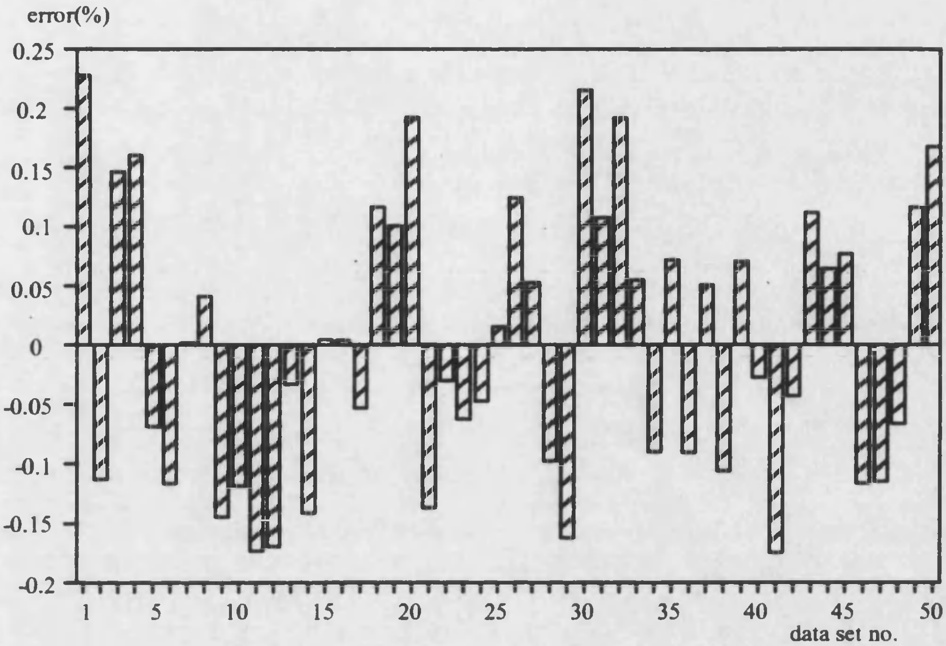


Figure 5.7 Low pressure training errors for the simulation transmission rig.

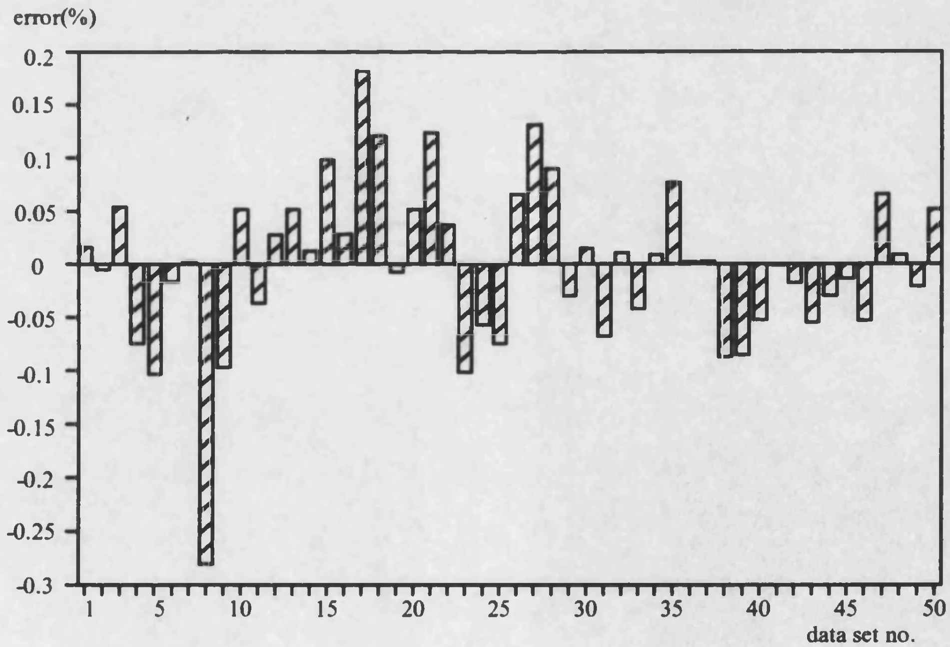


Figure 5.8 Flow rate training errors for the simulation transmission rig.

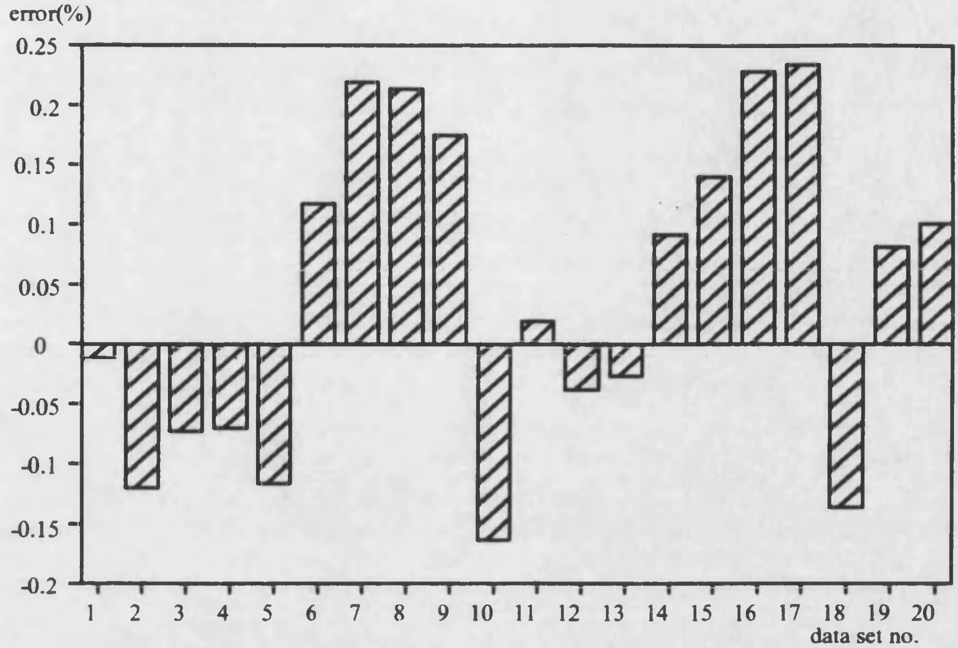


Figure 5.9 Pump torque prediction errors for the simulation transmission rig.

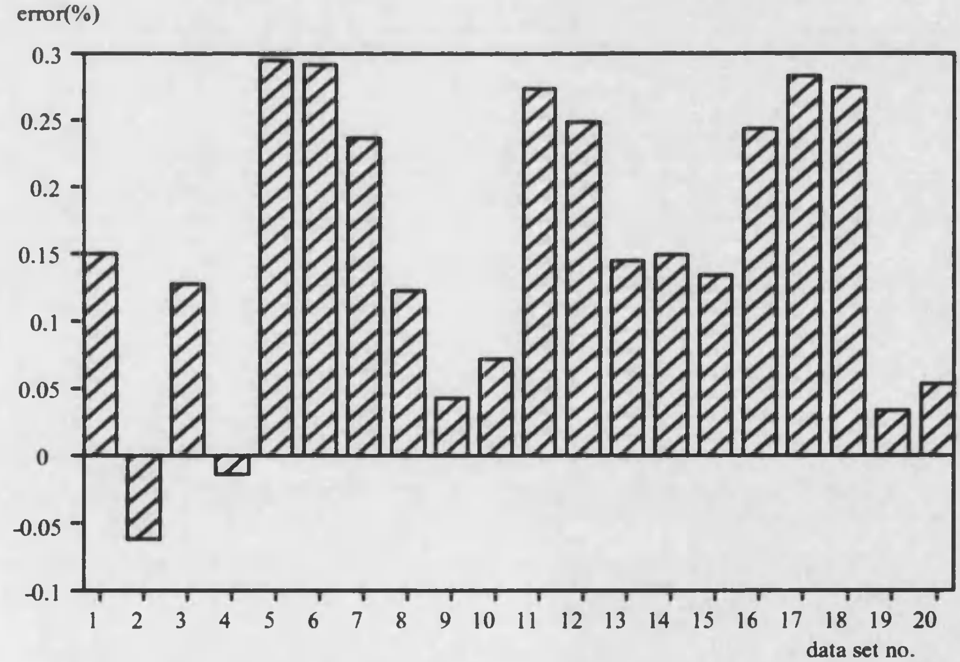


Figure 5.10 Motor torque prediction errors for the simulation transmission rig.

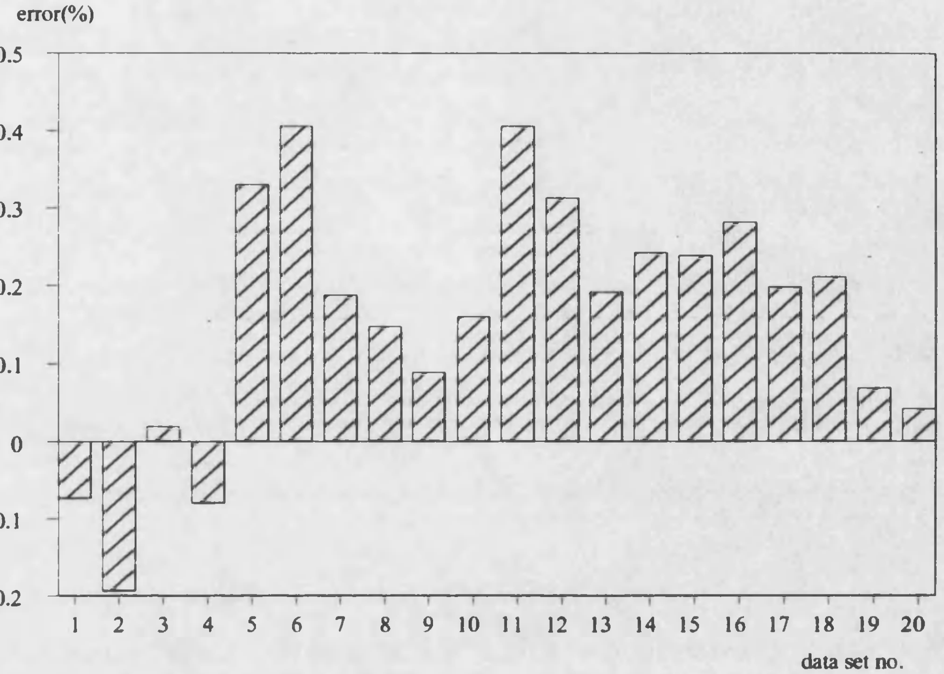


Figure 5.11 Motor speed prediction errors for the simulation transmission rig.

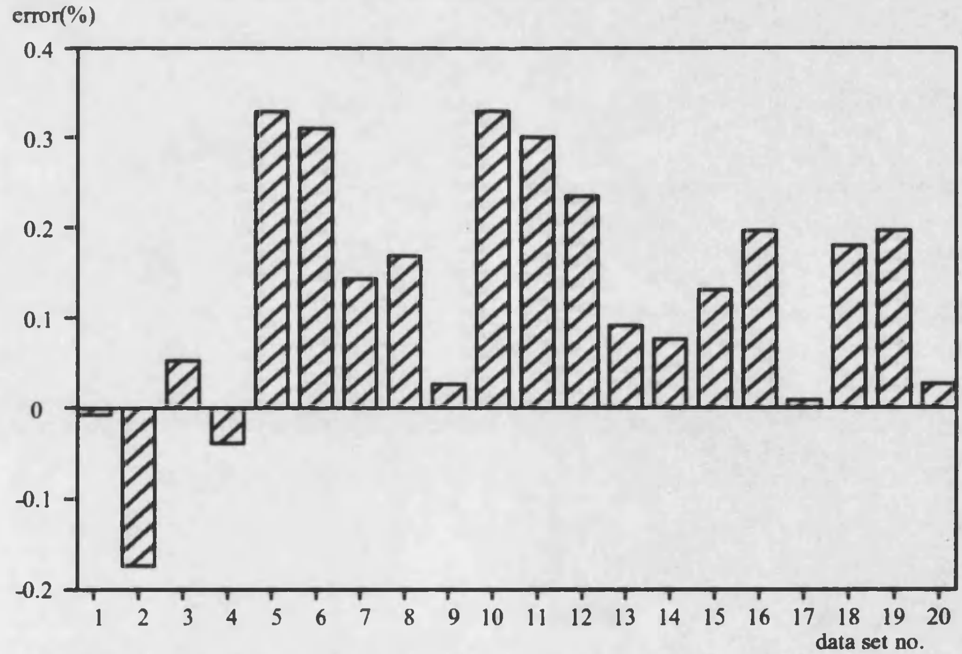


Figure 5.12 High pressure prediction errors for the simulation transmission rig.

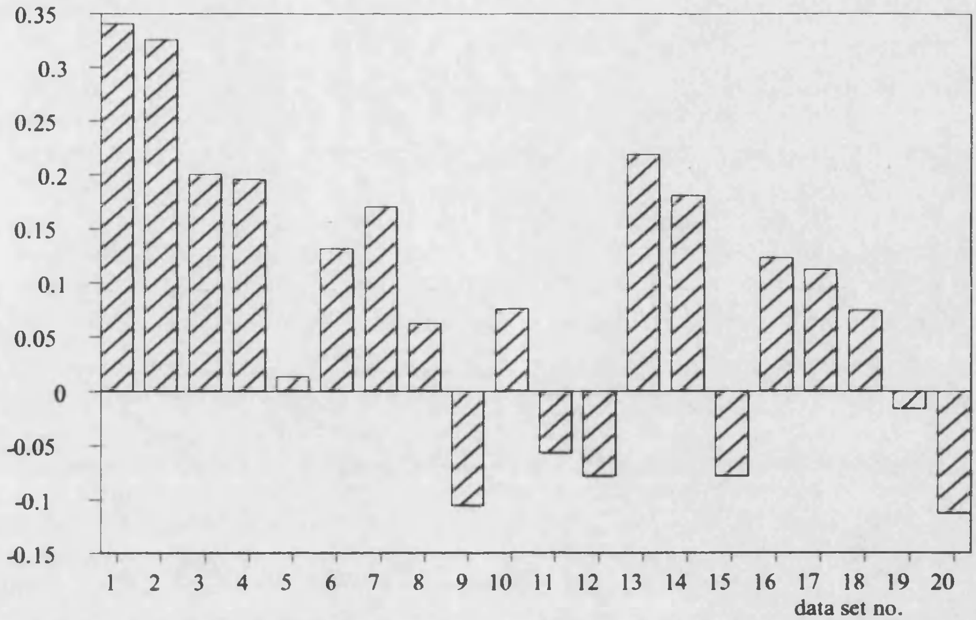


Figure 5.13 Low pressure prediction errors for the simulation transmission rig.

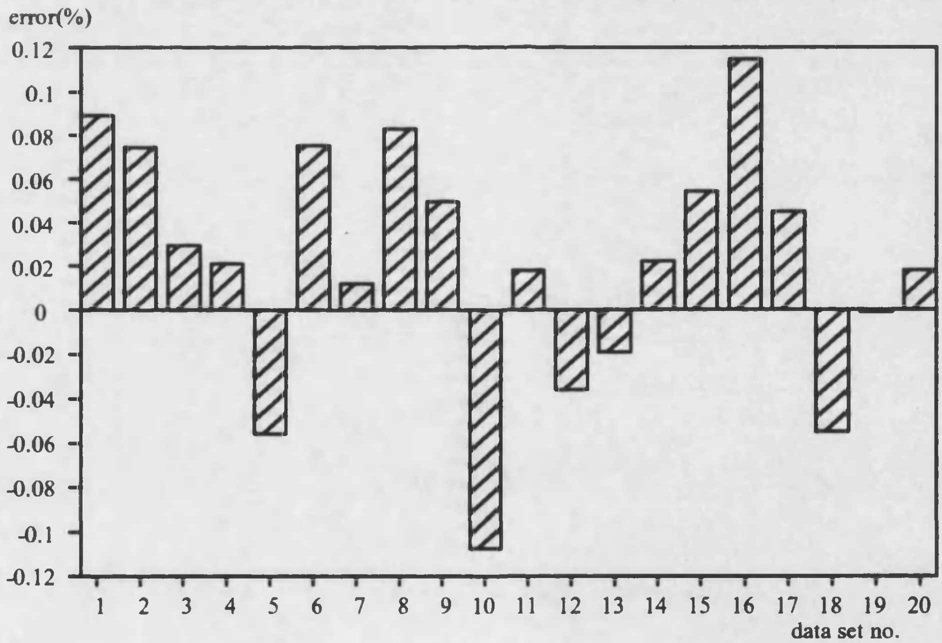
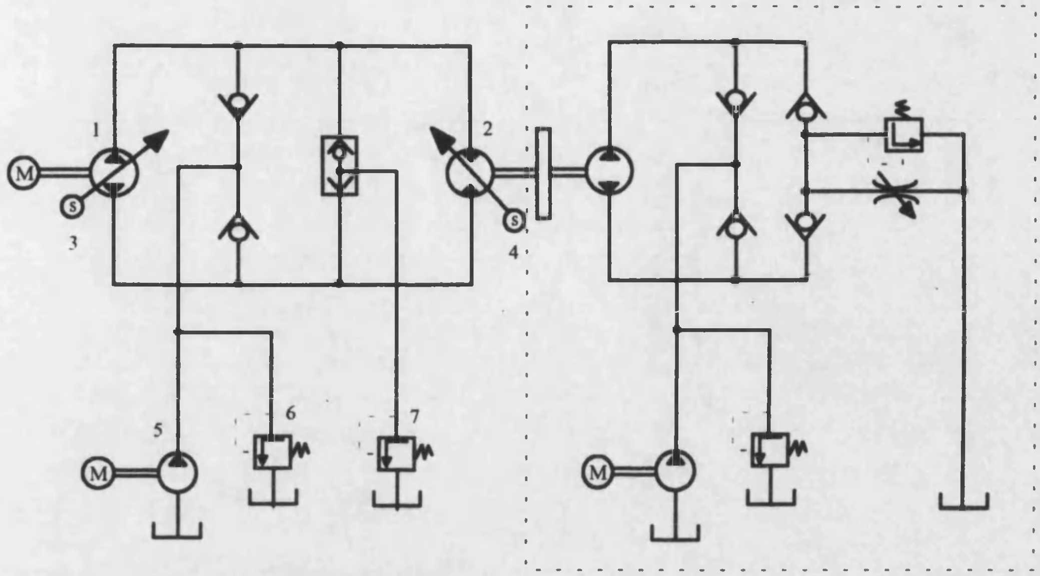


Figure 5.14 Flow rate prediction errors for the simulation transmission rig.



1 variable displacement piston pump

2 variable displacement piston motor

3 symbolized servo-system for pump swash-plate movement

4 symbolized servo-system for motor swash-plate movement

5 boost pump

6 low pressure line relief valve

7 high pressure line relief valve

8 load circuit(in the dot-line box)

Figure 5.15 Schematic diagram for the test transmission rig.

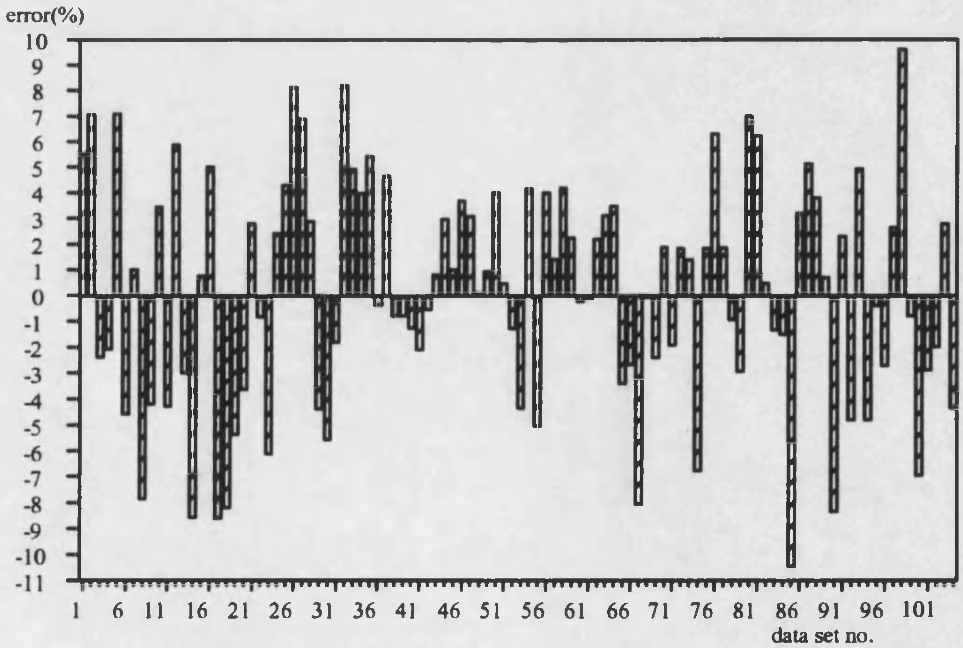


Figure 5.16 Pump torque training errors for the test transmission rig.

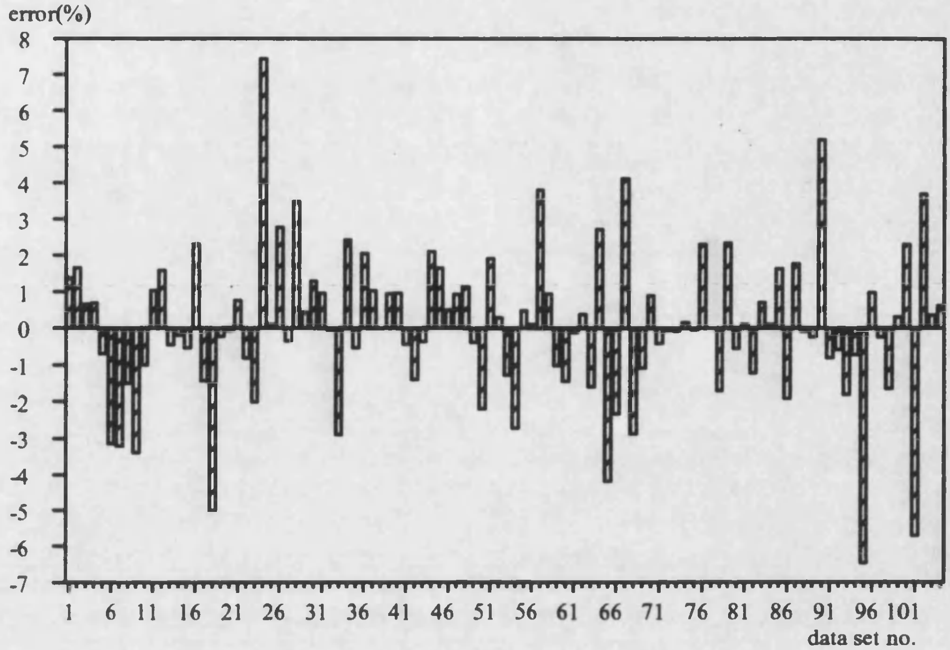


Figure 5.17 Motor torque training errors for the test transmission rig.

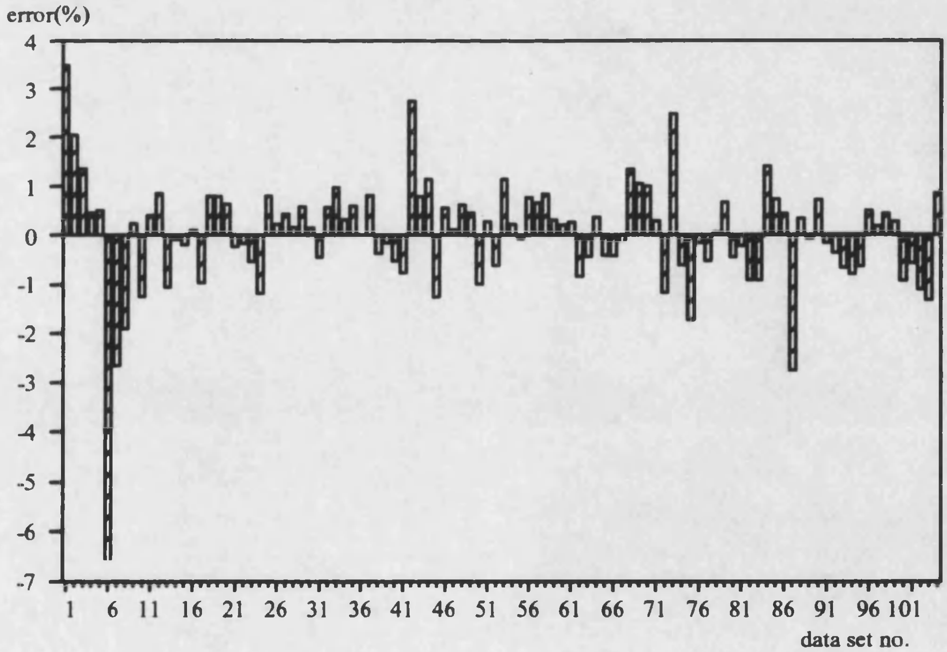


Figure 5.18 Motor speed training errors for the test transmission rig.

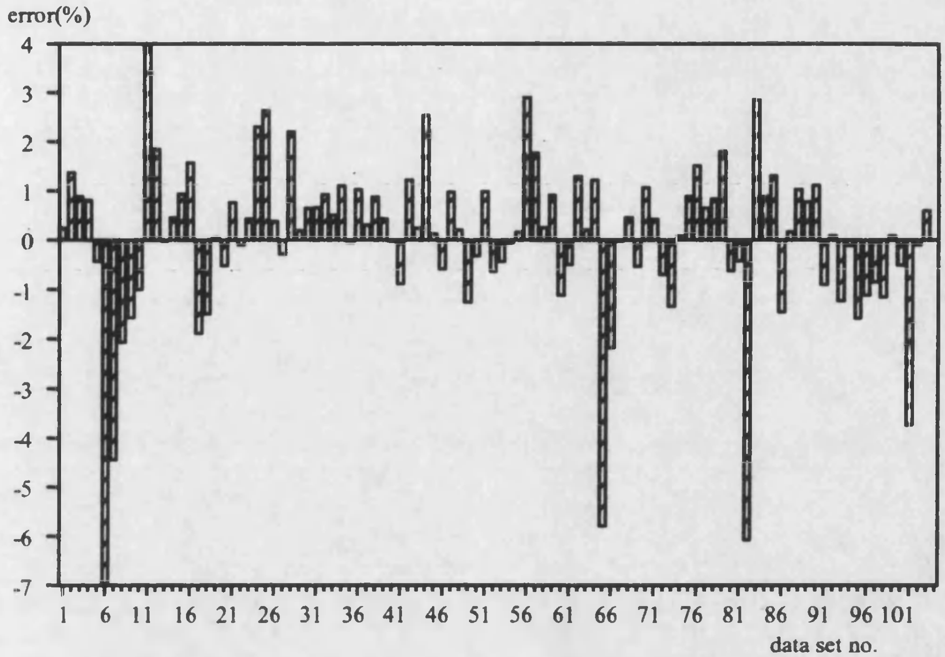


Figure 5.19 High pressure training errors for the test transmission rig.

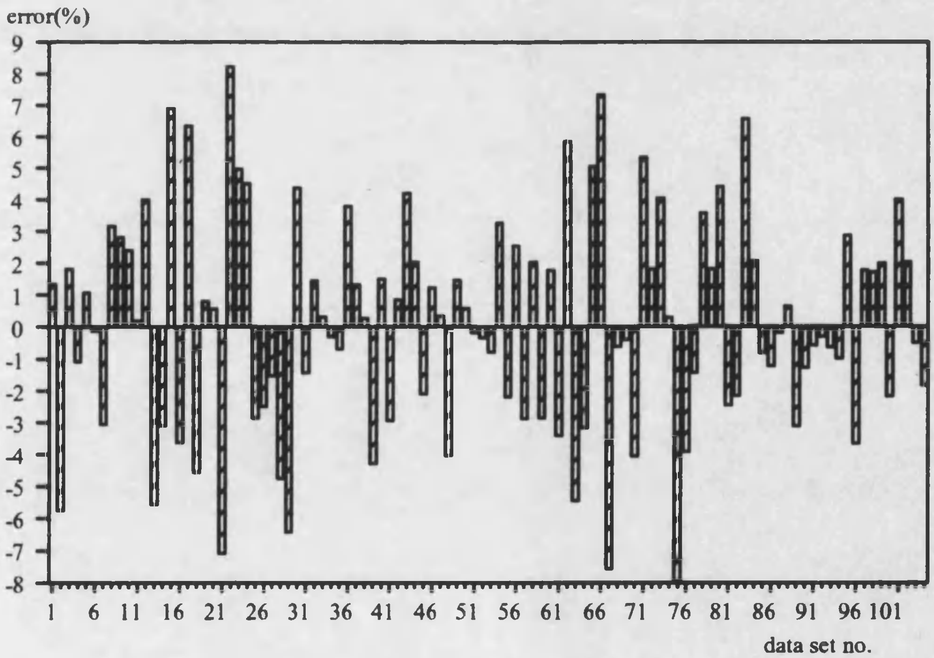


Figure 5.20 Low pressure training errors for the test transmission rig.

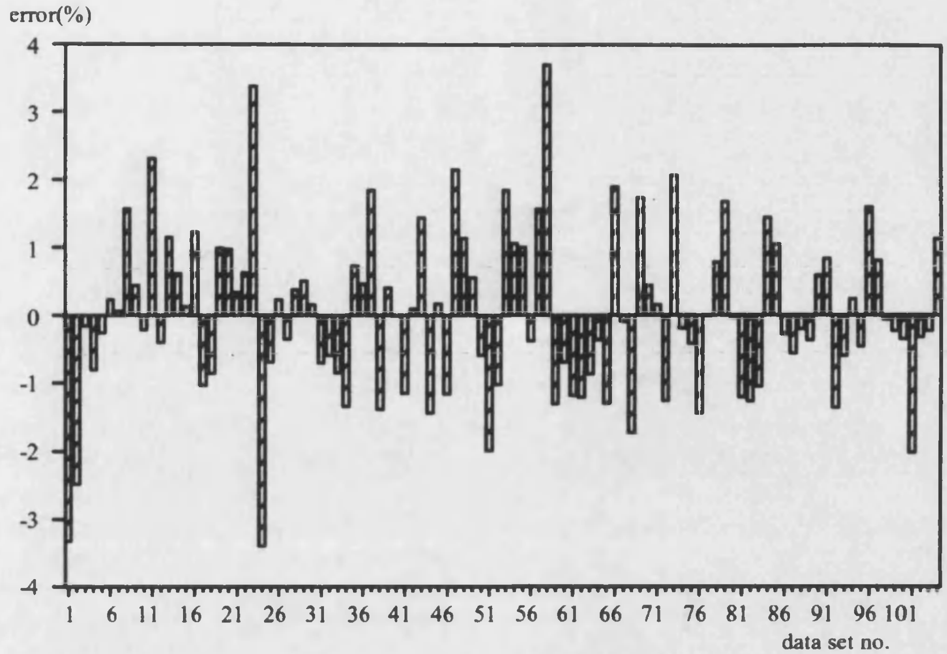


Figure 5.21 Flow rate training errors for the test transmission rig.

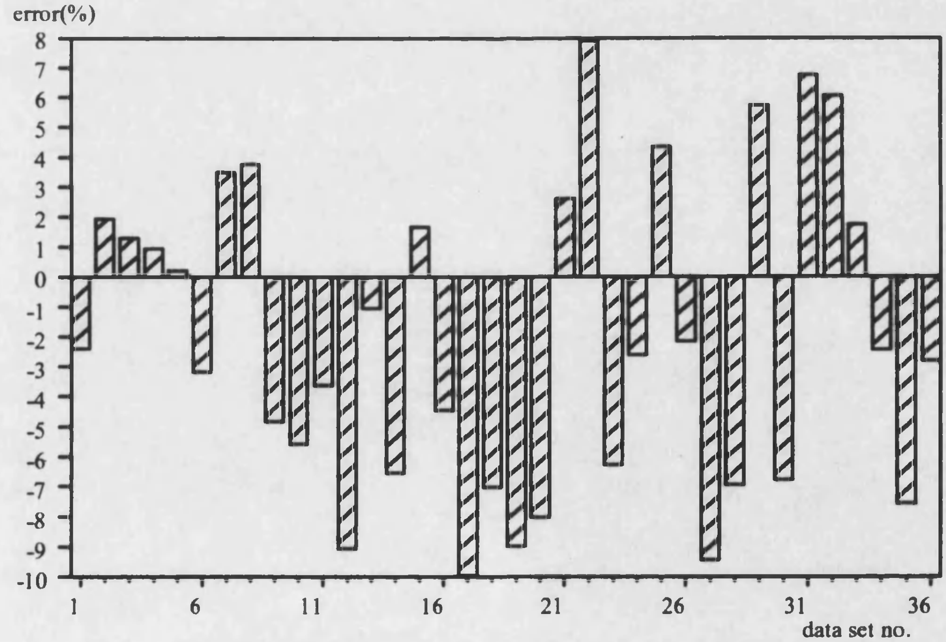


Figure 5.22 Pump torque prediction errors for the test transmission rig.

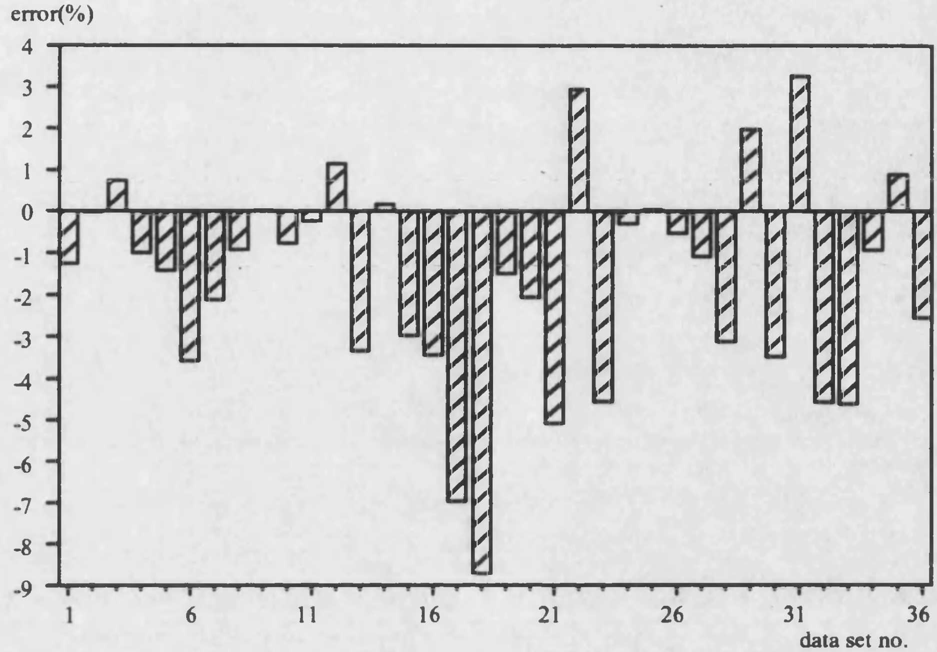


Figure 5.23 Motor torque prediction errors for the test transmission rig.

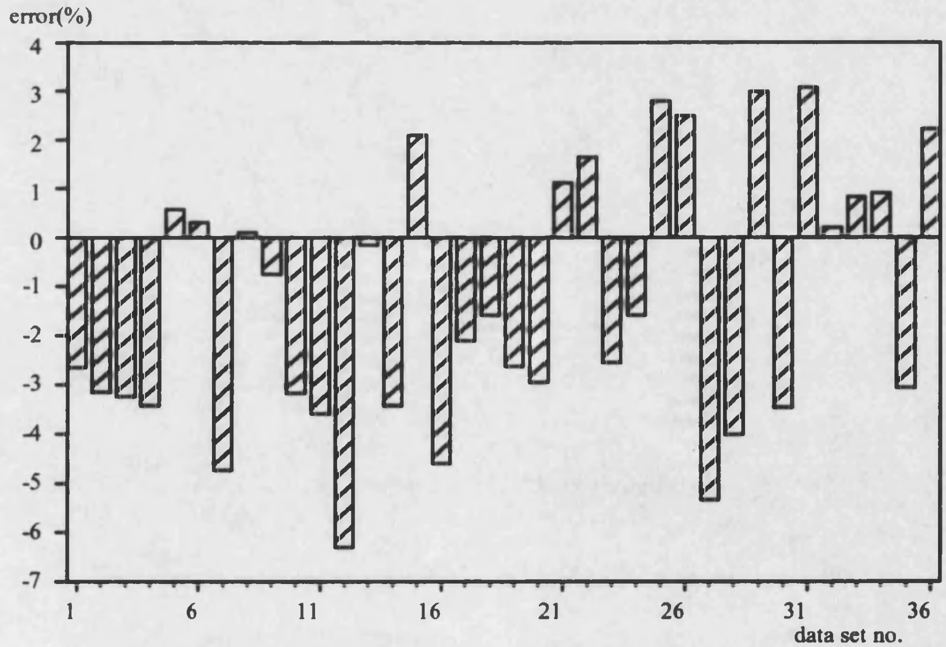


Figure 5.24 Motor speed prediction errors for the test transmission rig.

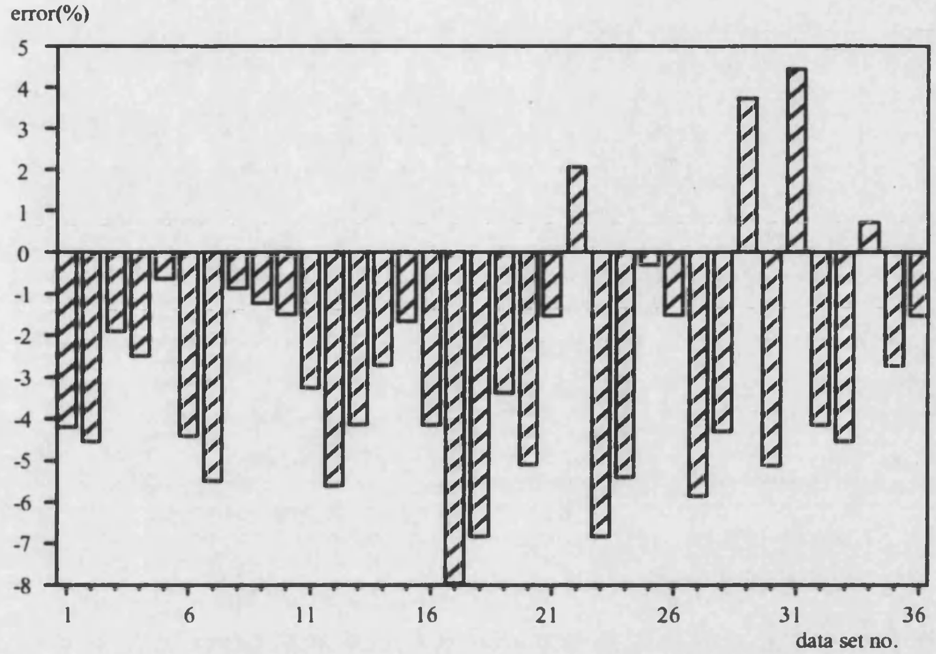


Figure 5.25 High pressure prediction errors for the test transmission rig.

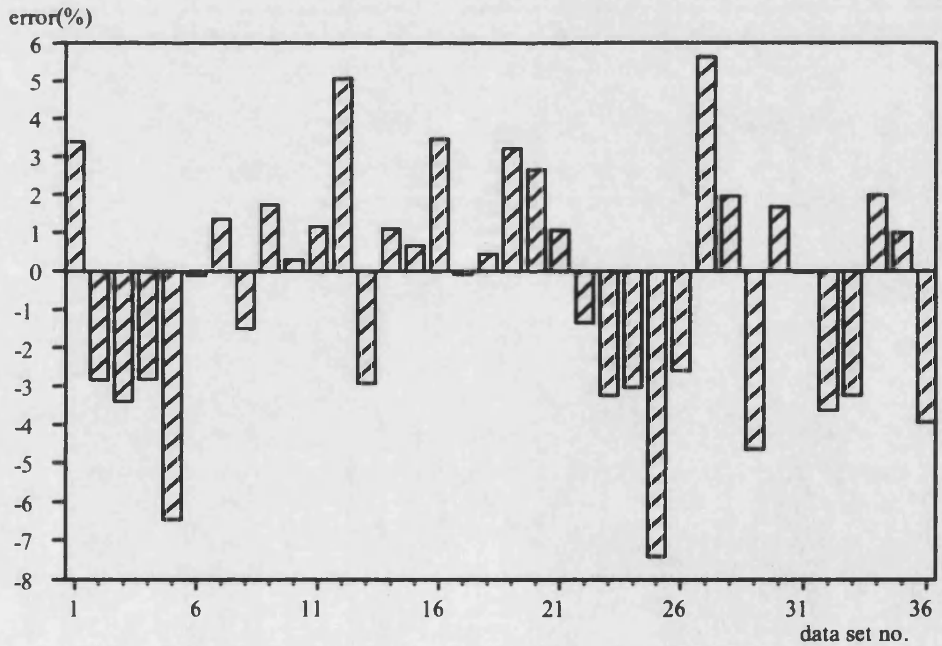


Figure 5.26 Low pressure prediction errors for the test transmission rig.

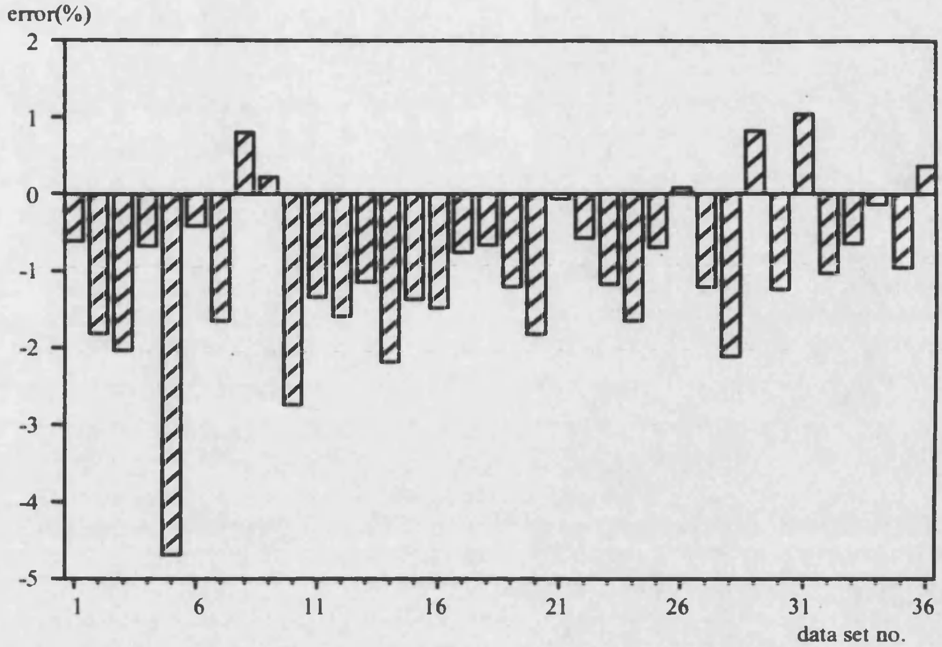


Figure 5.27 Flow rate prediction errors for the test transmission rig.

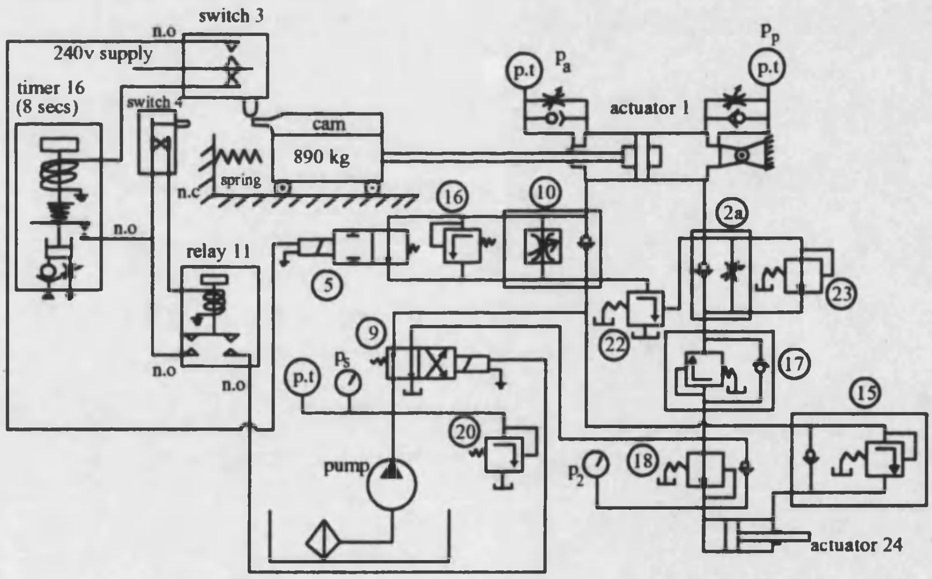


Figure 5.28 Test sequential circuit.

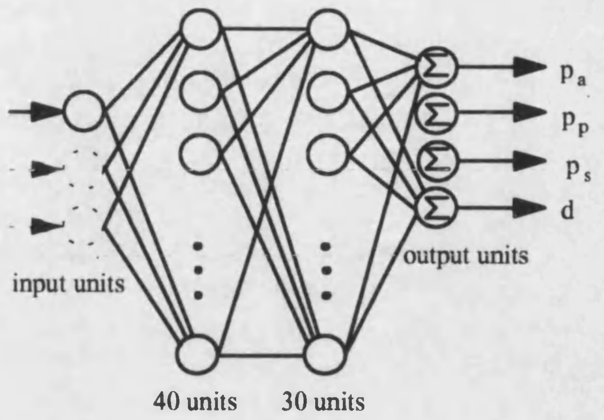


Figure 5.29 The structure of the neural model for the test seqential rig.

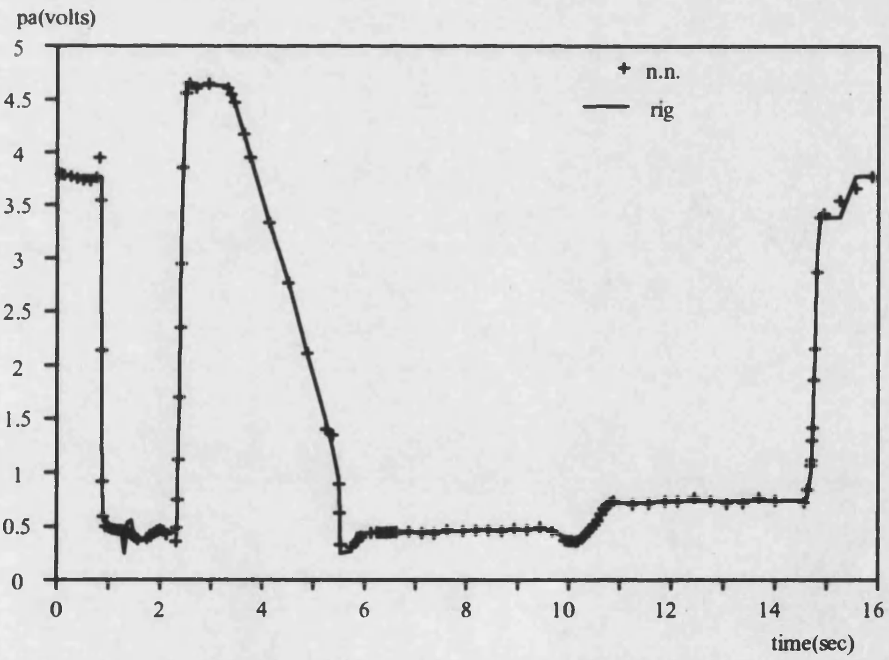


Figure 5.30 Time plot of the annulus pressure(pa) of the actuator(1) in the test rig and the pa outputs of the trained neural network.

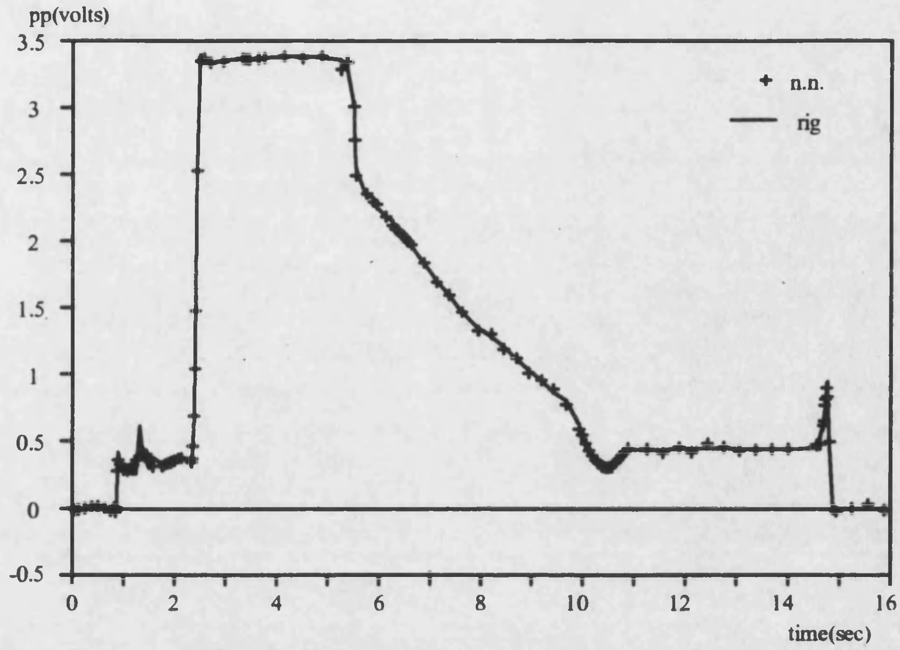


Figure 5.31 Time plot of the piston side pressure(pp) of the actuator(1) in the test rig and the pp outputs of the trained neural network.

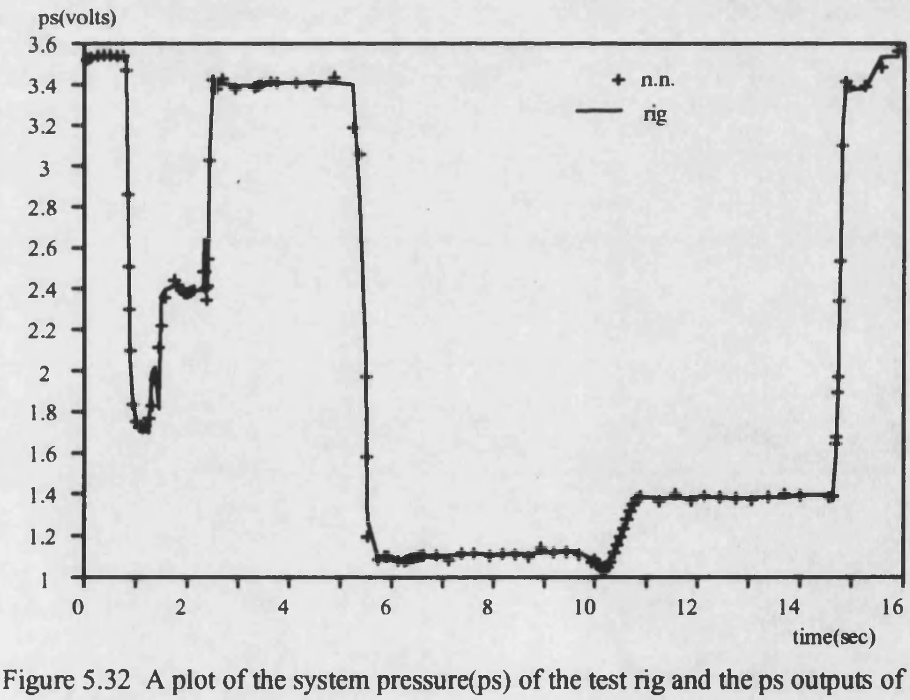


Figure 5.32 A plot of the system pressure(ps) of the test rig and the ps outputs of the trained neural network.

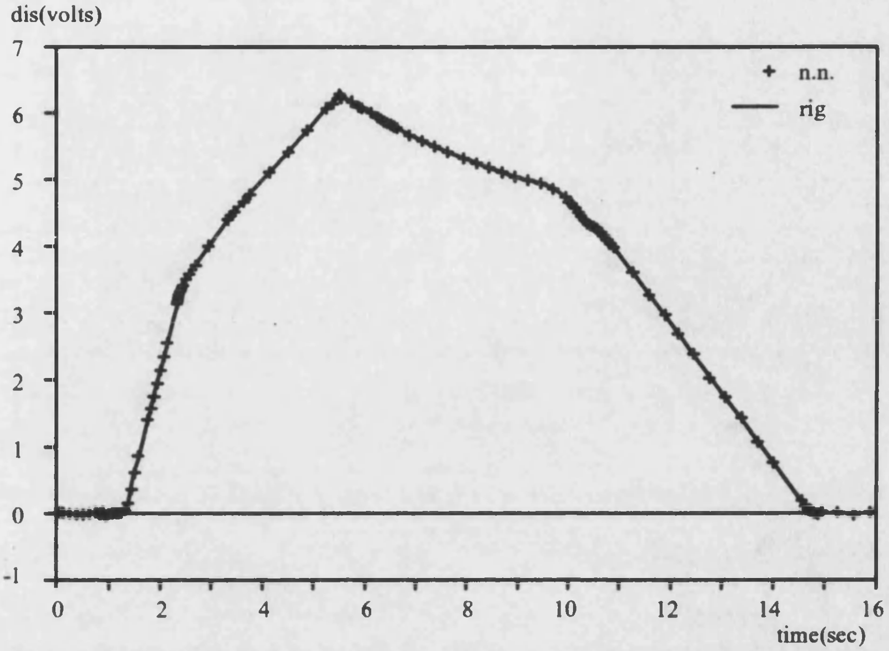


Figure 5.33 Time plot of the displacement(dis) of the actuator(1) in the test rig and the dis outputs of the trained neural network.

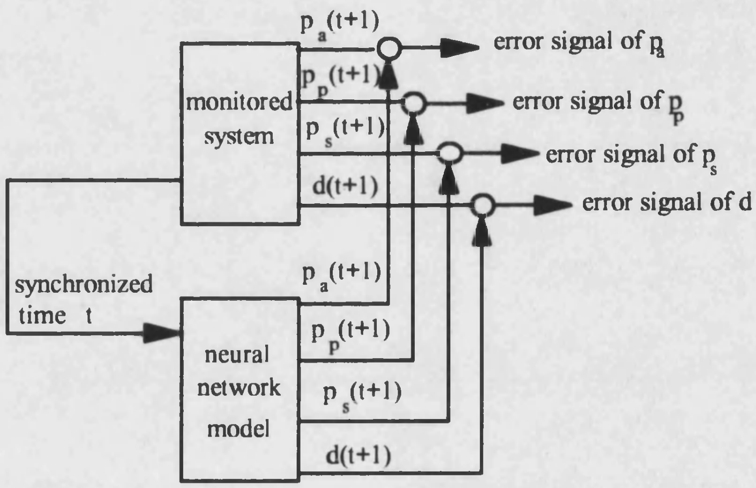


Figure 5.34 The schematic diagram of the monitor system using neural network as the reference model.

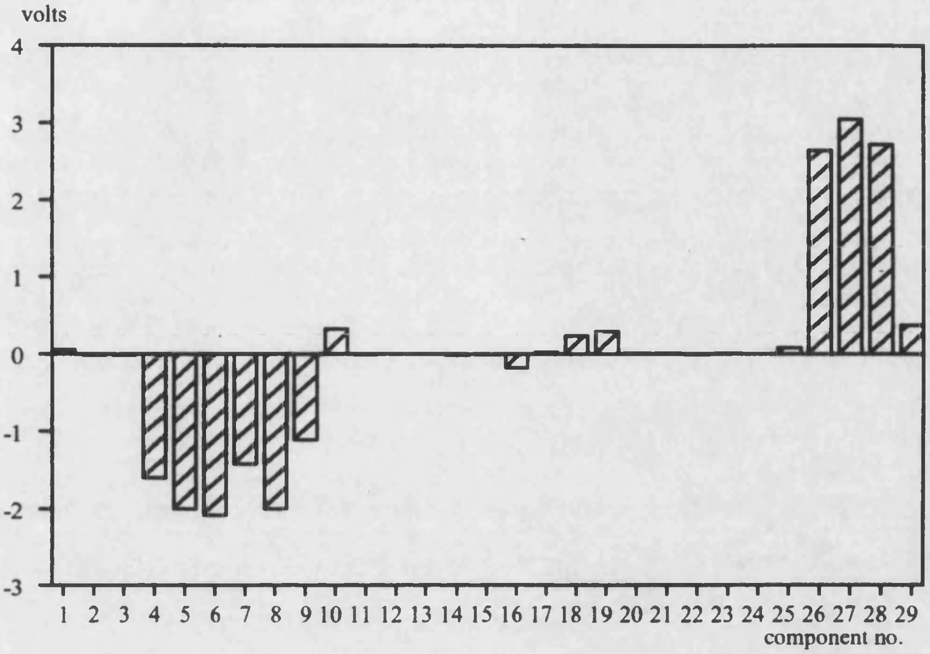


Figure 5.35 The annulus pressure error pattern of the fault class 1.

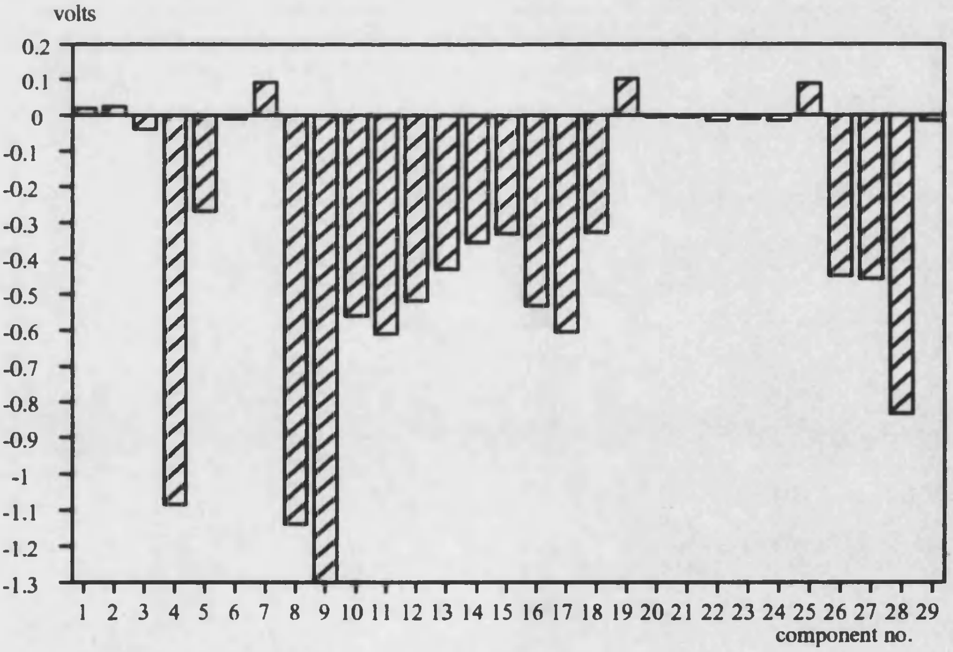


Figure 5.36 The piston end pressure error pattern of fault class 1.

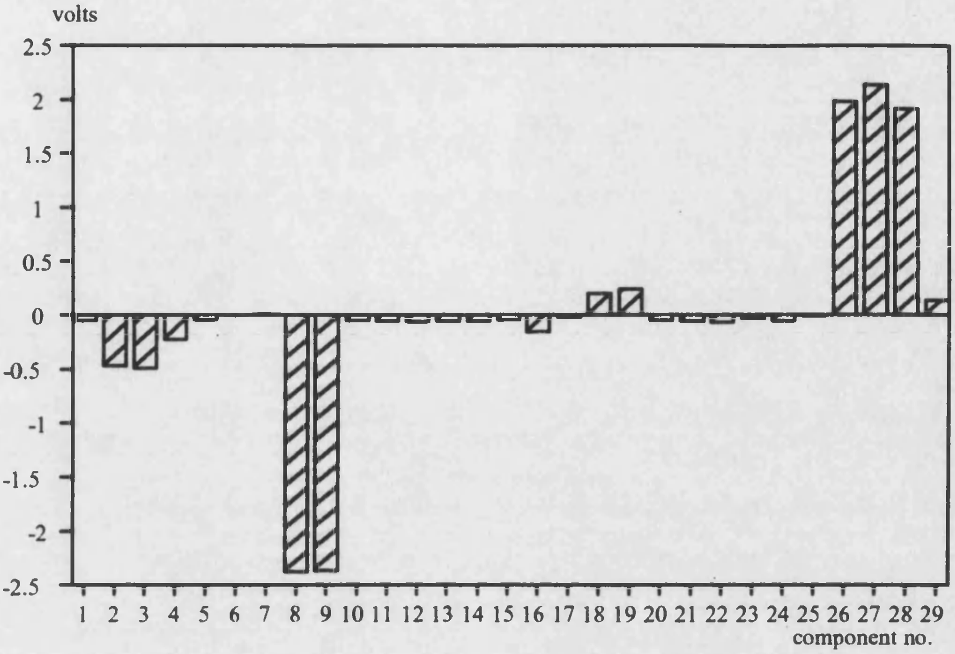


Figure 5.37 The system pressure error pattern of the fault class 1.

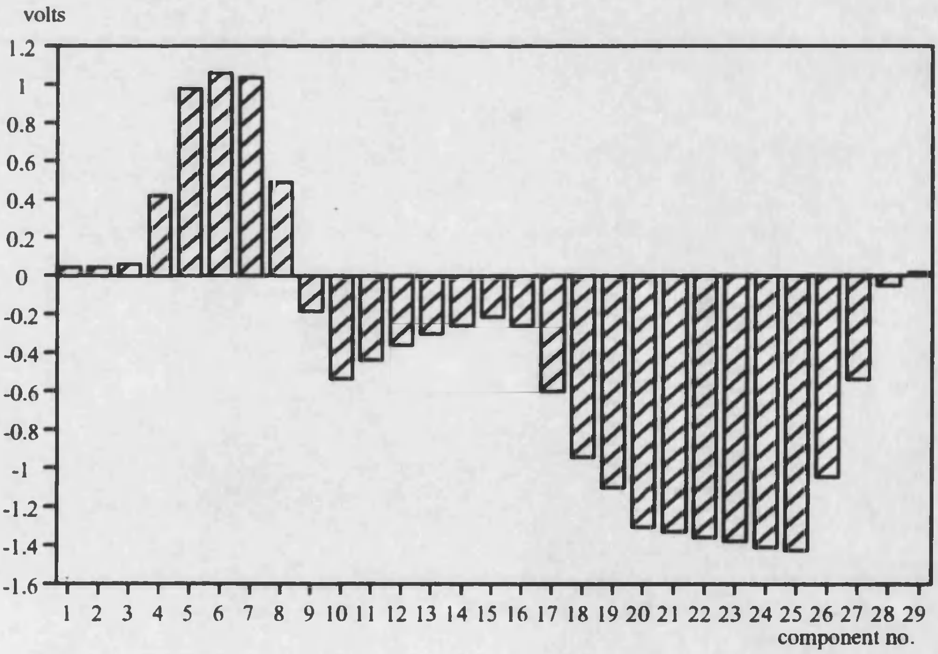


Figure 5.38 The displacement error pattern of the fault class 1.

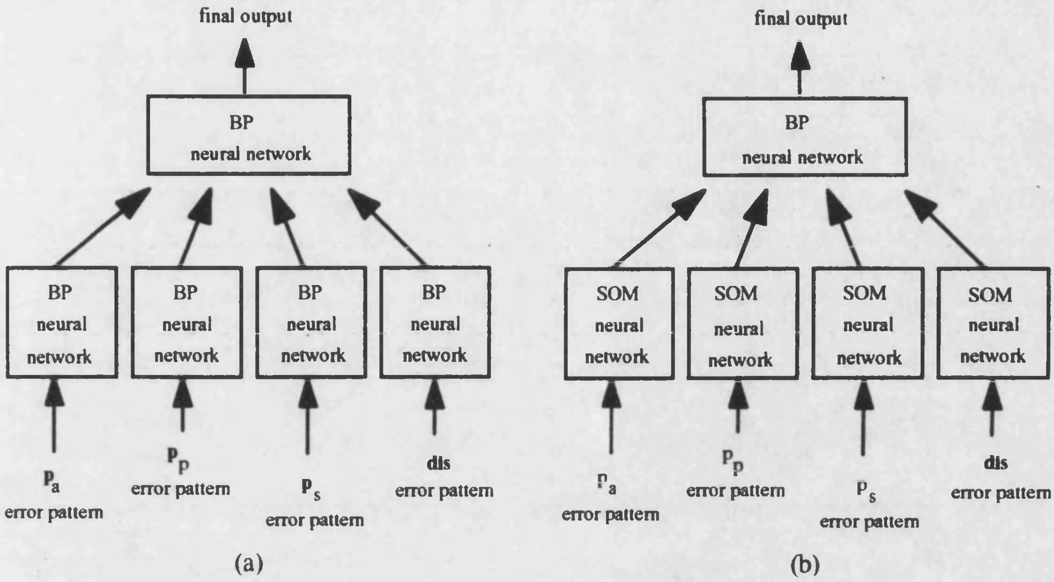


Figure 5.39 The hierarchical structures of the fault classifiers.

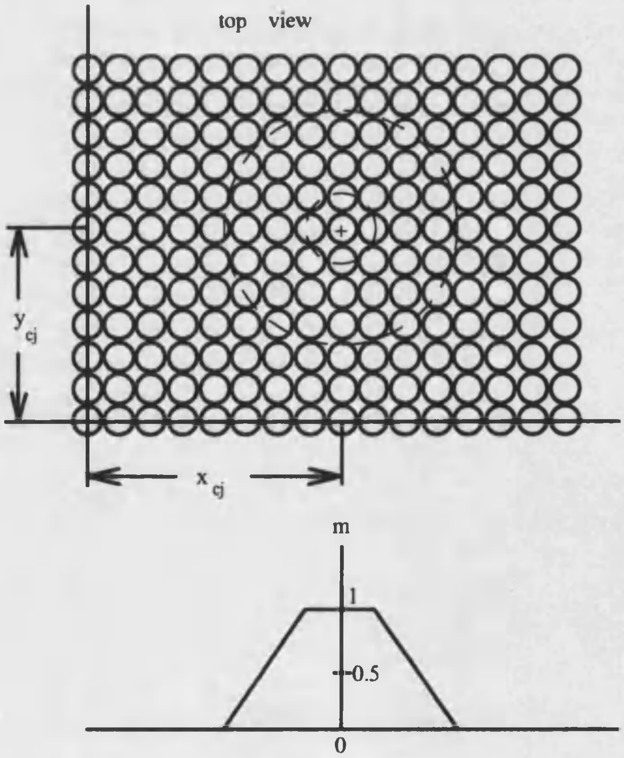


Figure 5.40 The membership function for the outputs of the SOM map.

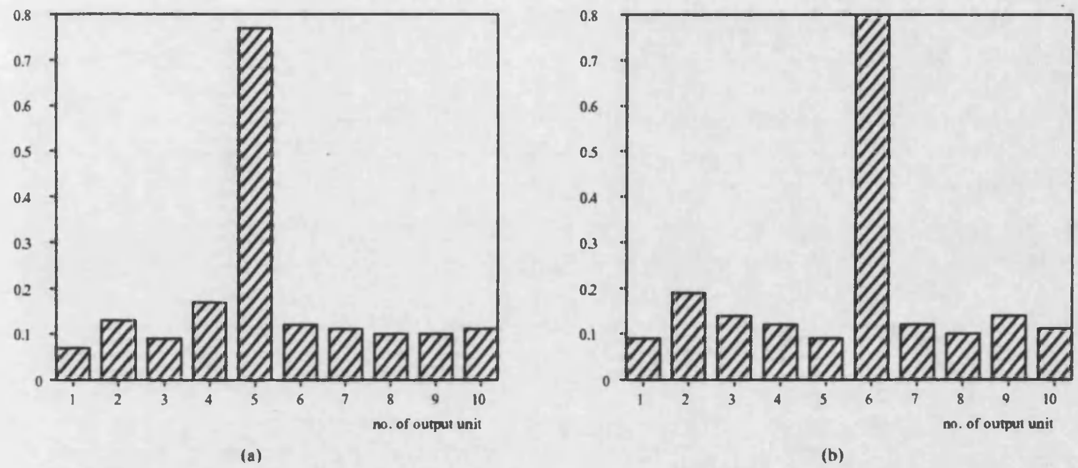


Figure 5.41 Two samples of the final outputs from the hierarchical structures shown in the Figure 5.39.

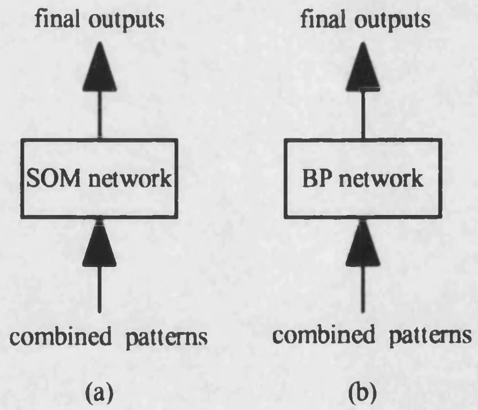


Figure 5.42 Alternative structures for classification test.

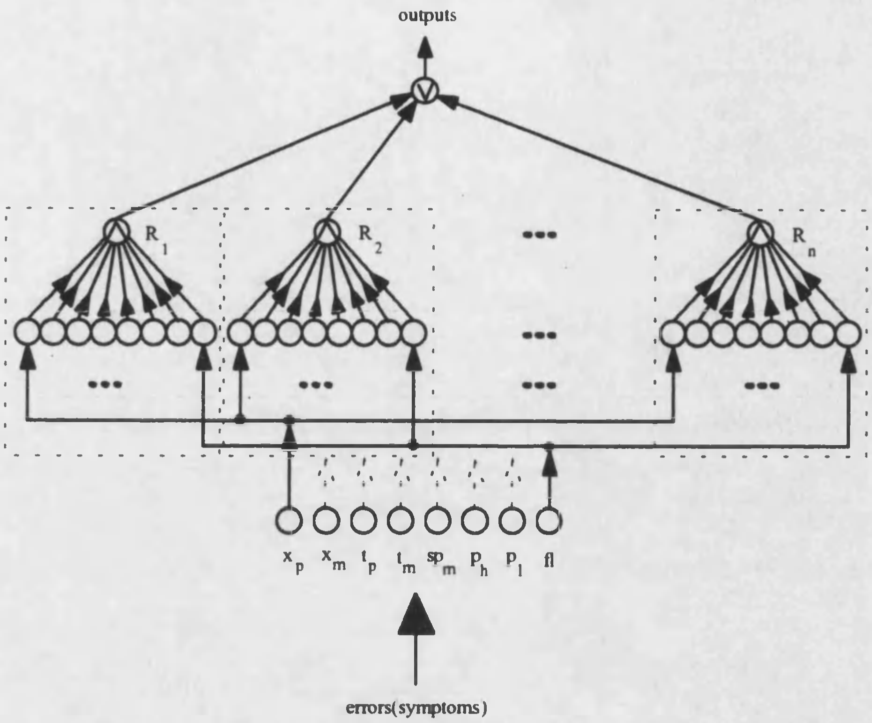


Figure 5.43 The computational network for the fuzzy inference mechanism applying min-operation on 'if' parts.

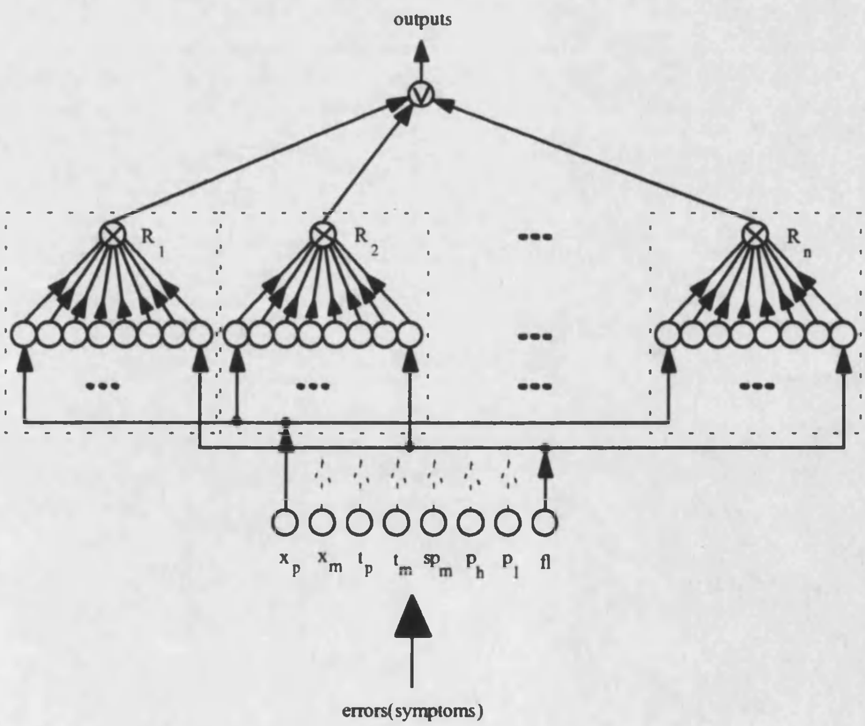


Figure 5.44 The computational network for the fuzzy inference mechanism applying product operation on 'if' parts.

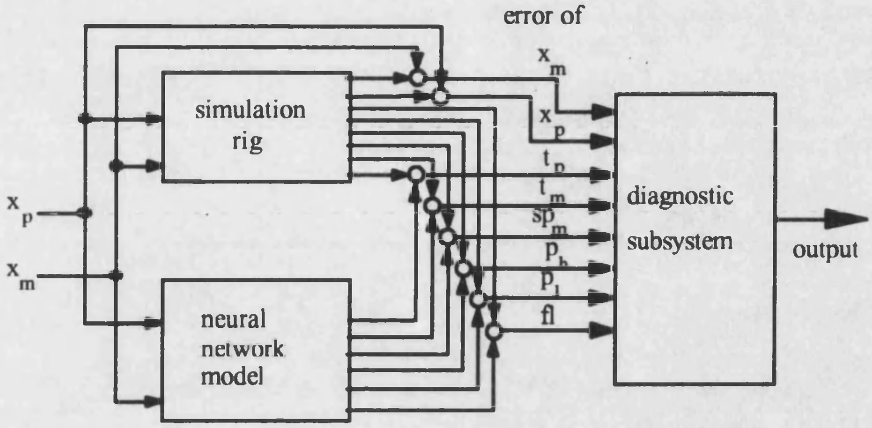


Figure 5.45 The skechmatic diagram of the monitoring system proposed for the simulation transmission rig.

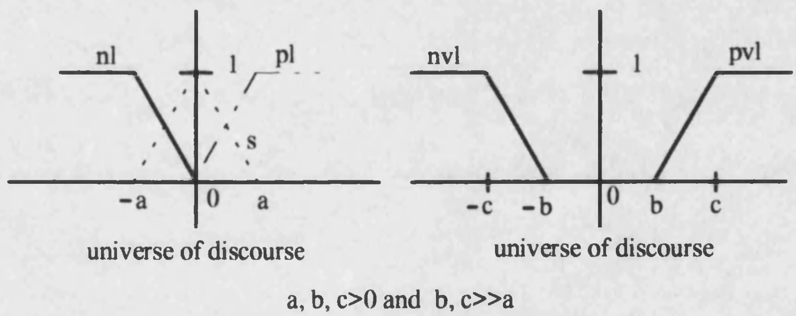


Figure 5.46 Membership functions for μ_Z(error), where z is one of the fuzzy set s, nl, pl, nvl, and pvl.

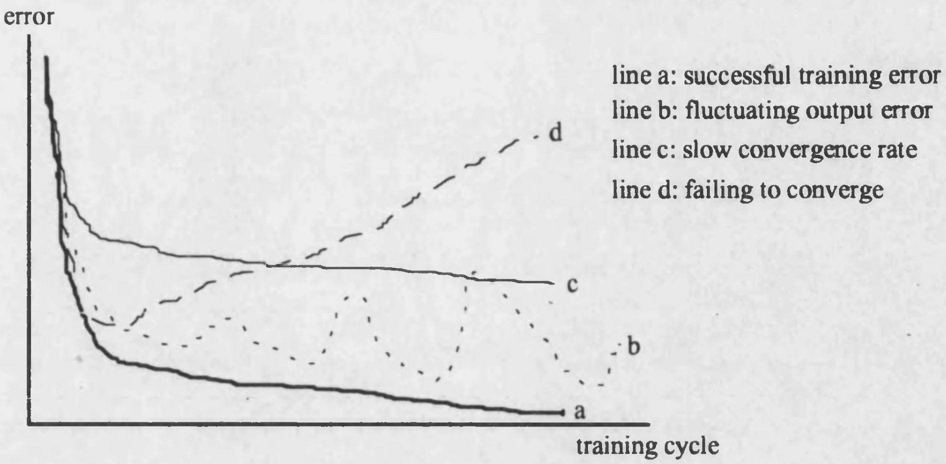


Figure 5.47 Schematic diagram of types of training errors.

Chapter 6
Conclusions and Recommendations for Future Research

6.1 Conclusions

In this thesis, three artificial neural networks, the back-propagation multi-perceptron network, the Kohonen's self-organising map, and the adaptive resonance theory network, were introduced by means of easily understood examples. All programs needed for this research were programmed using 'C' language and no commercial artificial neural networks programs were employed. By writing programs for these different neural networks it was possible to comprehensively understand the characteristics of variant neural networks, and easy to investigate the effect of parameters or structural changes on performance. Consequently, by changing program codes it was possible to develop new architectures of artificial neural networks, and link them to other software.

In order to detect faults, it is often necessary to accurately model a system. In this work, modelling techniques based on artificial neural networks were investigated and tested on two oil hydraulic test rigs and a simulation of the transmission system. BP neural networks were used to model these systems and the test results showed that the BP neural network modelling technique is very promising, not only because of the accuracy of the outputs predicted by the neural network models, but also because the models can be trained using real system data. Having trained the artificial neural network, the models can be used to detect possible discrepancies in the outputs of the monitored systems, and thereafter the faulty components, which cause the discrepancies, can be identified.

A principal aim of condition monitoring systems, in addition to detecting faults, is to diagnose faults or identify classes of faults. In this research two diagnostic techniques, derived from pattern recognition and from fuzzy logic, were developed and tested. With the pattern recognition technique, three different types of artificial neural networks, including back-propagation multi-perceptron and adaptive resonance theory networks, and self-organising map, were used. This technique was tested using data sets from the sequential test rig and the results generated a high percentage of correct classification rates. The performance of the ART2 network was better than the others. The ART2 network also has the advantage of being capable of automatically adopting new fault patterns. Hence, this network is recommended for fault classification. The technique derived from fuzzy logic was tested using a simulation of the transmission rig for both fault detection and classification simultaneously. A neural-like computational network was also proposed for this test. The results showed high correct detection and classification rates(about 92%). Theoretically, further improvements can be achieved by careful adjustment of the membership functions used in the rules. In conclusion, the evidence of these test results suggested that both pattern recognition and fuzzy logic classification techniques can be applied to fault diagnosis of fluid power systems.

Fuzzy logic is designed specifically for coping with information having uncertainty or ambiguity in its meaning and fuzzy logic systems can produce inferences by using approximating reasoning processes derived from fuzzy theories. The reasoning processes are close to the way in which decision are made by human beings. Accordingly, these systems are particularly suitable for situations which either heavily depend on human decisions or have ill-defined variables. In a condition monitoring system there are always some uncertainties in the diagnostic process and experts' experiences are often the main source of decision making criteria. Thus, fuzzy logic systems are considered to be one of the most powerful means of condition monitoring. Even though artificial neural network enforced pattern recognition techniques have proven to be effective for fault classification, monitoring schemes based on combined fuzzy logic and artificial neural networks are considered to have an even brighter future than techniques based solely on artificial neural networks. This area of research has been investigated in this thesis. It is shown that fuzzy logic systems can utilise both numerical and linguistic information and transform this information from linguistic form into a numerical form which is acceptable to physical systems. In addition, fuzzy logic systems do not need lengthy training or learning sessions, unlike almost all techniques using artificial neural networks in isolation. In terms of development, fuzzy logic systems can easily adopt newly acquired information, which can be described linguistically, by just adding new if-then rules to the rule base and they do not have the plasticity-stability problem existing in most artificial neural networks.

A literature survey into the use of artificial neural networks in the condition monitoring of fluid power systems revealed little work (only two have been reported in recent years[33,34]). For all of these studies artificial neural networks were limited to one type, namely BP networks. However, in this research three different types of artificial networks were used and compared, and different hierarchical structures of networks were suggested for fault diagnosis. No work on condition monitoring techniques which integrated fuzzy logic and the artificial neural network technique was found in the literature. In conclusion, it is believed that this research into the condition monitoring fluid power system has made several contributions by introducing novel, reliable and easily applicable condition monitoring schemes.

6.2 Recommendations for future research

The sequential rig used in this research suffered the problem of multiple faults classified by the same fault settings. This could be investigated further and extended by adding more transducers to differentiate between faults. In addition, a sequential circuit monitoring system based on fuzzy logic could be investigated.

In this research all the monitoring systems considered are used for detecting and diagnosing single faults and multiple fault cases are excluded. The probability of the unrelated multi-fault cases, which occur simultaneously at different positions and in unrelated machine components, are rather low, however multiple faults could occur due to the domino effect caused by a single faulty component. Hence, diagnosing multiple faults is a valuable subject for research. One possible approach to deal with multiple faults is to use fuzzy logic systems which have rules for detecting multiple faults.

The membership functions employed in the fuzzy diagnostic subsystems were set mainly by experience and partially by the sensitivity demand of detecting faults. An objective way to set up the membership functions is to use training methods to complete this task[86]. Thus, further research is suggested using experimental data and available if-then rules to train the neural networks to automatically adjust the parameters of the membership functions. Thereafter, if it is necessary, the membership functions can be adjusted manually according to the preferred detecting sensitivity.

Recent developments in the research of fuzzy neural networks have led to the publication of a number of new networks, for examples, fuzzy Kohonen clustering networks[94] which are derived from Kohonen's self-organising maps, adaptive-network-based fuzzy inference system(ANFIS)[87], back-propagation fuzzy system[86], fuzzy ARTMAP[96] which originates from the adaptive resonance theory networks, and the fuzzy min-max neural networks[97]. Following the preliminary studies presented here, it is argued that all of these networks are considered suitable for applying to the task of condition monitoring. Thus, further studies into the feasibility of directly applying these fuzzy neural networks to monitoring systems is recommended. These studies could be beneficial not only to the current subject but also to other related subjects, for instance, system identification and control.

References

[1] P. Hills (1987). Predictive maintenance, an economic alternative. Engineers' Digest, 26-29.

[2] C. Nolden (1987). Predictive maintenance. Plant Engineering, 41(4), 38-43.

[3] B.K.N. Rao (1990). Condition monitoring & diagnosis engineering management. Condition Monitoring and Diagnostic Technology, 1(2), 56-63.

[4] D. West (1991). Condition monitoring in manufacture. Condition Monitoring and Diagnostic Technology. 2(1), 3-6.

[5] R.A.Collacott(1977). Mechanical fault diagnosis. Chapman and Hall, London.

[6] L.F. Pau (1981). Failure diagnosis and performance monitoring. Marcel Dekker, New York.

[7] T.M. Hunt (1986). A review of condion monitoring techniques applicable to fluid power systems. 7th International Fluid Power Symposium, paper 33.

[8] R. Isermann (1984). Process fault detection based on modelling and estimation methods—a survey. *Automatica,* 20(4), 387-404.

[9] E. Czogala, J. Kopka and W. Pedrycz(1981). Diagnostics of systems in terms of fuzzy sets theory. Symposium on Technical Diagnostics, The Institute of Measurement and Control. 37-43.

[10] A.S. Willsky (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, 12, 601-611.

[11] R.K. Mehra and J. Peschon(1971). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7, 637-640.

[12] M. Basseville (1988). Detecting changes in signals and systems. *Automatica*, 24(3), 309-326.

[13] L.F. Pau (1986). Survey of expert systems for fault detection, test generation and maintenance. Expert Systems, 3(2), 100-111.

[14] R. Billington (1991). Application of expert systems for vibration based diagnosis. Condition Monitoring and Diagnostic Technology, 1(3), 99-101.

[15] R. Isermann and B. Freyermuth (1991). Process fault diagnosis based on process model knowledge (Part 1 and 2). J. of Dynamic Systems, Measurement, and Control, 103, 620-633.

[16] C.B. Scruby and R. Colbrook (1992). Novel applications of NDT to the monitoring of manufacturing processes. Condition Monitoring and Diagnostic Technology, 3(1), 2-8.

[17] K-A Elijah (1982). Acoustic emission sensing of tool wear in metal cutting: a general overview and problems. Proceedings of 10th NAMR Conference, 55-59.

[18] S. Bagnoli, R. Capitani and P. Citti(1988). Comparison of accelerometer and acoustic emission signals as diagnostic tools in assessing bearing damage. 2nd International Conference on Condition Monitoring, BHRA, 117-125.

[19] J. Darling, D.G. Tilley, D.N. Johnston, and C. Mu (1991). The use of acoustic emission for condition monitoring in high-pressure hydraulic pumps. Eurotech '91 Machine Systems, 137-140.

[20] T.M. Hunt, D.G. Tilley and C.G. Rice (1984). Techniques for the assessment of contamination in hydraulic oils. IMechE, C244/84.

[21] I. Raw and T. Hunt(1988). On-line debris monitoring-the way ahead. 2nd International Conference on Condition Monitoring, BHRA, 29-40.

[22] A. Prakash and O.P. Gandhi (1991). Identification of hydraulic system malfunctions using ferrography. Condition Monitoring and Diagnostic Technology, 2(2), 58-63.

[23] D.P. Anderson (1979). Ferrographic analysis for hydraulic fluids. SAE/SP-79/447.

[24] P.J. McClullagh (1984). Ferrography and particle analysis in hydraulic fluid power system. IMechE, C254/84.

[25] T.M. Hunt (1990). Oil debris monitoring - present and future problems. Condition Monitoring and Diagnostic Technology, 1(1), 13-17.

[26] W. Backe and T. Schwarz (1989). Condition monitoring on hydraulic displacement units. Second Bath International Fluid Power Workshop on Components and systems, Sept. 1989, 1-13.

[27] G.W. Chen (1988). Performance tests of electo-hydraulic servo valves and expert diagnosis system. MSc thesis, Chung-Cheng Institute of Technology, Taiwan, ROC. (in Chinese)

[28] J. Watton and D.J. Creber (1988). Leakage flow detection in a fluid power speed control system using a personal-computer-based condition monitoring package. Proc Instn Mech Engrs, 202(b4), 213-226.

[29] L.M. Wang (1990). An on-line computer-based monitoring system for hydrostatic transmissions. Fluid Power Centre, University of Bath, report No.1022.

[30] L.M. Wang (1990). Modelling , simulation and qualitative description of hydrostatic transmission performance. Fluid Power Centre, University of Bath, report No.1080.

[31] R.M. Atkinson, M.R. Montakhab, K.D.A. Pillay, D.J. Woollons, P.A. Hogan, C.R. Burrows and K.A. Edge(1992). Automated fault analysis for hydraulic systems, part 1: fundamentals. Proc Instn Mech Engrs, 206(I4), 207-214.

[32] P.A. Hogan, C.R. Burrows, K.A. Edge, R.M. Atkinson, M.R. Montakhab and D.J. Woollons(1992). Automated fault analysis for hydraulic systems, part 2: applications. Proc Instn Mech Engrs, 206(I4), 215-224.

[33] X.F. Lu, R. Burton and G. Schoenau (1994). Feasibility study on the use of a neural network to detect and locate excess piston wear in an axial piston pump. 7th Bath International Fluid Power Workshop on Innovations in Fluid Power.

[34] S. Daley and H. Wang(1993). On the application of neural networks to the monitoring of a simulated hydraulic rotary drive system. 6th Bath International Fluid Power Workshop on Innovations in Fluid Power.

[35] A.R. Baker and C.G. Windsor(1989). The classification of defects from ultrasonic data using neural networks: the Hopfield method. NDT International, 22(2), 97-105.

[36] S.J. Song and L.W. Schmerr(1992). Ultrasonic flaw classification in weldments using probabilistic neural networks. Journal of Nondestructive Evaluation, 11(2), 69-77.

[37] S. Rangwala and D. Dornfeld(1990). Sensor integration using neural networks for intelligent tool condition monitoring. ASME Journal of Engineering for Industry, 112, 219-228.

[38] S.V. Kamarthi, G.S. Sankar, P.H. Cohen and S.R.T. Kumara(1991). On-line tool wear monitoring using a Kohonen feature map. Proceedings of the Artificial Neural Networks in Engineering Conference, ASME, 639-644.

[39] M. Guillot and A.E. Ouafi(1991). On-line identification of tool breakage in metal cutting processes by use of neural networks. Proceedings of the Artificial Neural Networks in Engineering Conference, ASME, 701-709.

[40] D.A. Dornfield(1990). Neural network sensor fusion for tool condition monitoring. Annals of the CIRP, 39(1), 101-105.

[41] A.C. Okafor, M. Marcus and R. Tipirneni(1991). Multiple sensor integration via neural networks for estimating surface roughness and bore tolerance in circular end milling- part 1: time domain. Condition Monitoring and Diagnostic Technology, 2(2), 49-57

[42] S. Aylward, L. Sheckler, R. Anderson and J. Urnes(1991). Application of neural networks to in-flight aircraft diagnostics. Proceedings of the Artificial Neural Networks in Engineering Conference, ASME, 633-668.

[43] G.A. Carpenter(1989). Neural network models for pattern recognition and associative memory. Neural Networks, 2, 243-257.

[44] D.S. Levine(1991). Introduction to neural & cognitive modelling. Lawrence Erlbaum Associates, London. Chapter 2.

[45] P.K. Simpson(1990). Artificial neural systems: foundations, paradigms, applications, and implementations. Pergamon Press, Oxford.

[46] R.P. Lippmann(1987). An introduction to computing with neural nets. IEEE ASSP Magazine, April, 4-22.

[47] B.Widrow and M. Lehr(1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proceedings of the IEEE, 78(9), 1415-1442.

[48] M.M. Nelson and W.T. Illingworth(1991). A practical guide to neural nets Addison-Wesley, New York, page 30.

[49] K. Fukushima(1983). Neocognition: a neural network model for a mechanism of visual pattern recognition. IEEE Transactions, SMC-13, 826-834.

[50] G.A. Carpenter and S. Grossberg(1987). A massively parallel architecture for a self-organising neural pattern recognition machine. Computer vision, Graphics, and Image Processing, 37, 54-115.

[51] G.A. Carpenter and S. Grossberg(1987). ART 2: self-organisation of stable category recognition codes for analogue input patterns. Applied Optics, 26, 4919-4930.

[52] G.A. Carpenter and S. Grossberg(1990). ART 3: hierarchical search using chemical transmitters in self-organising pattern recognition architecture. Neural Networks, 3, 129-152.

[53] J. Hopfield(1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79, 2554-2558.

[54] J. Hopfield(1984). Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of the National Academy of Sciences, 81, 3088-3092.

[55] T. Kohonen(1990). The self-organising map. Proceedings of the IEEE, 78(9), 1464-1477.

[56] T. Kohonen(1988). The "neural" phonetic typewriter. IEEE Computer, 21, 11-22.

[57] E. Davalo and P. Naim(1991). Neural networks. Mamillan Education, London, 72-80.

[58] R. Hecht-Nielsen(1987). Counterpropagation networks. Applied Optics, 26, 4979-4984.

[59] D.E. Rumelhart, G.E. Hinton, and R.J. Williams(1986). Learning internal representations by error propagation. In Parallel Distributed Processing, volume 1, MIT Press, Cambridge MA., 318-362.

[60] J.A. Freeman and D.M. Skapura(1991). Neural networks: algorithms, applications and programming techniques. Addison-Wesley, Reading MA.

[61] S. Brunak and B. Lautrup(1988). Neural networks: computers with intuition. World Scientific, Copenhagen.

[62] K.J. Hunt, D. Sbarbaro, R.Zbikowski and P.J. Gawthrop(1992). Neural networks for control systems—a survey. *Automatica*, 28(6), 1083-1112.

[63] D.E. Rumelhart, G.E. Hinton, and R.J. Williams(1986). A general framework for parallel distributed processing. In Parallel Distributed Processing, volume 1, MIT Press, Cambridge MA., 45-76.

[64] B. Kosko(1992). Neural networks and fuzzy systems. Prentice-Hall, New Jersey.

[65] S-I. Amari(1983). Field theory of self-organising neural nets. IEEE Transactions SMC-13, 741-748.

[66] V. Vemuri(1988). Artificial neural networks: an introduction. In Artificial Neural Networks: Theoretical Concepts. V. Vemuri ed., 1-12.

[67] R. Hecht-Nielsen(1988). Neurocomputer applications. Neural Computers, NATO ASI Series, F41, 444-453.

[68] H.G. Schuster(1992). Applications of neural networks. VCH Publishers, New York.

[69] C.H. Dagli, A.R.T. Kumara and Y.C. Shin ed.(1991). Intelligent engineering systems through artificial neural networks. ASME Press, New York.

[70] P.G.J. Lisboa ed.(1992). Neural networks current applications. Chapman & Hall, London.

[71] J.C. Bezdek(1993) Editorial fuzzy models— what are they, and why? IEEE Transaction on Fuzzy Systems, 1(1), 1-6.

[72] C.C. Klimasauskas(1992). Neural networks: an engineering perspective. IEEE Communication Magazine, September, 50-53.

[73] B.Widrow and M. Lehr(1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proceedings of the IEEE, 78(9), 1415-1442.

[74] S.E. Fahlman(1988). Faster learning variations on back-propagation: an empirical study. Proceedings of the 1988 Connectionist Models Summer School, 38-51.

[75] R.S. Scalero and N. Tepedelenlioglu(1992). A fast new algorithm for training feedforward neural networks. IEEE Trans. on Signal Processing, 40(1), 202-210.

[76] J.K. Kruschke and J.R. Movellan(1991). Benefits of gain: speeded learning and minimal layers in back-propagation networks. IEEE Trans. SMC, 21, 273-280.

[77] K.S. Narendra and K. Parthasarathy(1990). Identification and control of dynamical systems using neural networks. IEEE Trans. on Neural Networks, 1(1), 4-27.

[78] J.A. Kangas, T. Kohonen and J.T. Leaksonen(1990). Variants of self-organizing maps. IEEE Trans. on Neural Networks, 1(1), 93-99.

[79] N.R Pal, J.C. Bezdek and E.C.-K. Tsao(1993). Generalized clustering networks and Kohonen's self-organizing scheme. IEEE Trans. on Neural Networks, 4(4), 549-557.

[80] L.A. Zadeh(1965). Fuzzy sets. Information and Control, 8, 338-353.

[81] L.A. Zadeh(1968). Fuzzy algorithms. Information and Control, 12, 94-102.

[82] L.A. Zadeh(1973). Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. SMC-3, 28-44.

[83] J. Maiers and Y.S. Sherif(1985). Applications of fuzzy set theory. IEEE Trans. SMC, 15(1), 175-189.

[84] Y.K. Young(1993). Fuzzy theory and applications.(in Chinese) Chuan Hua Science and Technology Book Co. Taipei, Taiwan, ROC.

[85] C.H. Wang(1993). Fuzzy technology.(in Chinese) Chuan Hua Science and Technology Book Co. Taipei, Taiwan, ROC.

[86] L-X Wang and J. M. Mendel(1992). Back-propagation fuzzy system as nonlinear dynamic system identifiers. Proc. IEEE Int. Conf. Fuzzzy Systems, 1409-1418.

[87] J-S R. Jang(1993) ANFIS: adaptive-networks-based fuzzy inference system. IEEE Trans. SMC, 23(3), 655-684.

[88] C-T Lin and C.S.G. Lee(1991). Neural-network-based fuzzy logic control and decision system. IEEE Trans. on Computer, 40(12), 1320-1336.

[89] C.C. Lee(1990). Fuzzy logic in control systems: fuzzy logic controller, part 1 and part 2. IEEE Trans. SMC, 20(2), 404-435.

[90] J.C. Bezdek(1993). Fuzzy models—what are they, and why? IEEE Trans. on Fuzzy systems, 1(1), 1-6.

[91] S.K. Pal and S. Mitra(1992). Multilayer perceptron, fuzzy sets, and classification. IEEE Trans. on Neural Networks, 3(5), 683-697.

[92] J.M. Keller and H. Tahani(1992). Backpropagation neural networks for fuzzy logic. Information Science, 62, 205-221.

[93] H. Ishibuchi, R. Fujioka and H. Tanaka(1993). Neural networks that learn from fuzzy if-then rules. IEEE trans. on Fuzzy Systems, 1(2), 85-97.

[94] J.C. Bezdek, E.C-K Tsao and N.R. Pal(1992). Fuzzy Kohonen clustering networks. Proc. IEEE Int. Conf. Fuzzzy Systems, 1035-1041.

[95] G.A. Carpenter, S. Grossberg and D.B. Rosen(1991). Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 4, 759-771.

[96] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds and D.B. Rosen(1992). Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. on Neural Networks, 3(5), 698-713.

[97] P.K. Simpson(1992). Fuzzy min-max neural networks—part 1: Classification. IEEE Trans. on Neural Networks, 3(5), 776-786.

[98] P.K. Simpson(1993). Fuzzy min-max neural networks—part 2: Clustering. IEEE Trans. on Fuzzy Systems, 1(1), 32-45.

[99] J.M. Keller, R.R. Yager and H. Tahani(1992). Neural network implementation of fuzzy logic. Fuzzy Sets and Systems, 45, 1-12.

[100] A.B. Bulsari, A. Krastawski and H. Saxen(1992). Implementing a fuzzy expert system in an artificial neural network. European Symposium on Computer Aided Process Engineering -2, s405-s410.

[101] J.C. Bezdek(1992). Computing with uncertainty. IEEE Communication Magazine, 24-36.

[102] BATH*fp* simulation package users' manuals, Fluid Power Centre, University of Bath.

[103] Design of hydraulic systems(H2 course notes), Fluid Power Centre, University of Bath.

[104] A.B. Bulsari, A. Krastawski and H. Saxen(1992). Implementing a fuzzy expert system in an artificial neural network. European Symposium on Computer Aided Process Engineering-2, s405-s410.

Appendix A Training and prediction results of the simulation transmission rig

Training data sets for the simulation transmission rig and the trained neural network outputs with their errors in percentages are listed as follows.(The errors are plotted in Figures 5.3 to Figure 5.8.)

Sample no. 1
Input data: 1.0000 1.0000
Target: 10.3120 1.9900 14.8960 7.9480 1.0620 8.3690
Output: 10.3231 1.9920 14.8984 7.9524 1.0644 8.3703
err(%): 0.1074 0.1002 0.0162 0.0551 0.2282 0.0161
Sample no. 2
Input data: 1.0000 0.7500
Target: 13.4280 2.4790 19.7880 11.3930 1.0620 8.3540
Output: 13.4221 2.4814 19.7960 11.3960 1.0608 8.3536
err(%): -0.0440 0.0956 0.0402 0.0265 -0.1124 -0.0053
Sample no. 3
Input data: 1.0000 0.5000
Target: 19.5090 3.1240 26.2280 18.1050 1.0520 8.3240
Output: 19.5975 3.1193 26.1821 18.0718 1.0535 8.3285
err(%): 0.4535 -0.1497 -0.1749 -0.1832 0.1465 0.0542
Sample no. 4
Input data: 0.7500 1.0000
Target: 6.8530 1.6170 11.1710 6.3610 1.0620 6.2760
Output: 6.8318 1.6159 11.1572 6.3561 1.0637 6.2713
err(%): -0.3090 -0.0693 -0.1240 -0.0767 0.1606 -0.0751
Sample no. 5
Input data: 0.7500 0.7500
Target: 8.6400 1.9840 14.8410 8.9830 1.0620 6.2650
Output: 8.6225 1.9806 14.8040 8.9681 1.0613 6.2586
err(%): -0.2030 -0.1706 -0.2494 -0.1654 -0.0683 -0.1029
Sample no. 6
Input data: 0.7500 0.5000
Target: 13.1100 2.7050 22.0510 15.5410 1.0620 6.2360
Output: 13.0972 2.7022 22.0114 15.5228 1.0608 6.2350
err(%): -0.0978 -0.1033 -0.1798 -0.1169 -0.1159 -0.0163
Sample no. 7
Input data: 0.5000 1.0000
Target: 4.1030 1.2460 7.4460 4.7800 1.0620 4.1830
Output: 4.0751 1.2472 7.4497 4.7810 1.0620 4.1831
err(%): -0.6805 0.0977 0.0503 0.0210 0.0021 0.0026
Sample no. 8
Input data: 0.5000 0.7500
Target: 4.9300 1.4890 9.8910 6.5830 1.0620 4.1750
Output: 4.9315 1.4890 9.8962 6.5930 1.0624 4.1633
err(%): 0.0312 -0.0006 0.0529 0.1526 0.0414 -0.2802
Sample no. 9
Input data: 0.5000 0.5000
Target: 6.9790 1.9700 14.7000 11.0480 1.0620 4.1560
Output: 6.9799 1.9633 14.6635 11.0363 1.0605 4.1520
err(%): 0.0131 -0.3390 -0.2486 -0.1062 -0.1447 -0.0964
Sample no. 10
Input data: 0.5000 0.6000
Target: 5.8890 1.7310 12.3110 8.6730 1.0620 4.1660
Output: 5.8777 1.7329 12.3007 8.6790 1.0607 4.1682
err(%): -0.1913 0.1083 -0.0837 0.0693 -0.1184 0.0522
Sample no. 11
Input data: 1.0000 0.9000
Target: 11.2920 2.1540 16.5320 9.0310 1.0620 8.3640
Output: 11.2792 2.1501 16.5282 9.0196 1.0602 8.3609
err(%): -0.1134 -0.1823 -0.0228 -0.1265 -0.1725 -0.0366
Sample no. 12
Input data: 1.0000 0.6000
Target: 17.0810 2.9620 24.6250 15.4310 1.0620 8.3360
Output: 17.1304 2.9602 24.6173 15.4229 1.0602 8.3383
err(%): 0.2892 -0.0591 -0.0314 -0.0527 -0.1696 0.0276
Sample no. 13
Input data: 1.0000 0.5500
Target: 18.9210 3.1800 26.8010 17.4660 1.0620 8.3270
Output: 18.6682 3.1678 26.6932 17.3989 1.0616 8.3313
err(%): -1.3361 -0.3849 -0.4023 -0.3844 -0.0330 0.0516
Sample no. 14
Input data: 0.8000 0.9000
Target: 8.1250 1.8230 13.2250 7.5550 1.0620 6.6910
Output: 8.1127 1.8201 13.2147 7.5520 1.0605 6.6918
err(%): -0.1510 -0.1592 -0.0779 -0.0398 -0.1416 0.0125
Sample no. 15
Input data: 0.8000 0.6500
Target: 10.9250 2.3220 18.2190 11.4110 1.0620 6.6740
Output: 10.9296 2.3204 18.2058 11.4047 1.0620 6.6806
err(%): 0.0421 -0.0677 -0.0724 -0.0556 0.0041 0.0986
Sample no. 16
Input data: 0.8000 0.5500
Target: 13.0610 2.6440 21.4450 14.3520 1.0620 6.6610
Output: 13.0722 2.6473 21.4694 14.3759 1.0620 6.6629
err(%): 0.0856 0.1256 0.1140 0.1664 0.0041 0.0283
Sample no. 17
Input data: 0.6000 0.8000
Target: 5.9760 1.6140 11.1410 6.9770 1.0620 5.0130
Output: 6.0039 1.6121 11.1319 6.9641 1.0614 5.0221
err(%): 0.4673 -0.1176 -0.0813 -0.1846 -0.0524 0.1817
Sample no. 18
Input data: 0.6000 0.6500
Target: 7.0980 1.8660 13.6630 9.0250 1.0620 5.0040
Output: 7.1019 1.8691 13.6833 9.0165 1.0632 5.0100
err(%): 0.0544 0.1675 0.1484 -0.0940 0.1164 0.1207
Sample no. 19
Input data: 0.6000 0.5500
Target: 8.3210 2.1080 16.0840 11.2580 1.0620 4.9950
Output: 8.3257 2.1084 16.0944 11.2606 1.0631 4.9946
err(%): 0.0565 0.0207 0.0649 0.0232 0.1004 -0.0078
Sample no. 20
Input data: 0.6000 0.6000
Target: 7.6420 1.9780 14.7760 10.0180 1.0620 5.0000
Output: 7.6439 1.9798 14.7978 10.0207 1.0640 5.0026
err(%): 0.0254 0.0894 0.1474 0.0268 0.1925 0.0516
Sample no. 21
Input data: 0.9000 1.0000
Target: 8.8430 1.8410 13.4060 7.3130 1.0620 7.5320
Output: 8.8600 1.8433 13.4242 7.3011 1.0605 7.5413
err(%): 0.1921 0.1268 0.1357 -0.1627 -0.1374 0.1235
Sample no. 22
Input data: 0.9000 0.7500
Target: 11.3830 2.2810 17.8100 10.4280 1.0620 7.5180
Output: 11.4010 2.2793 17.8124 10.4270 1.0617 7.5208
err(%): 0.1582 -0.0751 0.0136 -0.0092 -0.0299 0.0373
Sample no. 23
Input data: 0.9000 0.5000
Target: 17.6400 3.1220 26.2030 18.1000 1.0610 7.4840
Output: 17.5519 3.1022 26.0329 18.0073 1.0603 7.4764
err(%): -0.4997 -0.6339 -0.6491 -0.5124 -0.0625 -0.1015
Sample no. 24
Input data: 0.8000 1.0000
Target: 7.4870 1.6910 11.9160 6.6780 1.0620 6.6950
Output: 7.4653 1.6941 11.9364 6.6898 1.0615 6.6911
err(%): -0.2897 0.1832 0.1709 0.1773 -0.0478 -0.0576
Sample no. 25
Input data: 0.8000 0.8000
Target: 8.9740 1.9860 14.8560 8.7240 1.0620 6.6860
Output: 8.9675 1.9839 14.8252 8.7193 1.0622 6.6810
err(%): -0.0725 -0.1059 -0.2071 -0.0542 0.0155 -0.0748
Sample no. 26
Input data: 0.8000 0.5000
Target: 14.5790 2.8520 23.5200 16.4430 1.0620 6.6520
Output: 14.5620 2.8522 23.5375 16.4416 1.0633 6.6564
err(%): -0.1167 0.0075 0.0744 -0.0085 0.1242 0.0655
Sample no. 27
Input data: 0.6000 0.9000
Target: 5.4860 1.4920 9.9180 6.0830 1.0620 5.0170
Output: 5.5180 1.4903 9.9041 6.0669 1.0626 5.0236
err(%): 0.5841 -0.1157 -0.1403 -0.2650 0.0538 0.1306
Sample no. 28
Input data: 0.6000 0.7500
Target: 6.2850 1.6870 11.8720 7.5420 1.0620 5.0110
Output: 6.3055 1.6874 11.8745 7.5383 1.0610 5.0155
err(%): 0.3263 0.0245 0.0214 -0.0488 -0.0972 0.0897
Sample no. 29

Input data: 0.6000 0.5000
Target: 9.1890 2.2640 17.6420 12.8400 1.0620 4.9880
Output: 9.2023 2.2641 17.6354 12.8276 1.0603 4.9865
err(%): 0.1451 0.0037 -0.0375 -0.0966 -0.1617 -0.0301
Sample no. 30
Input data: 0.6000 0.6000
Target: 7.6420 1.9780 14.7760 10.0180 1.0620 5.0000
Output: 7.6440 1.9812 14.8087 10.0332 1.0643 5.0007
err(%): 0.0263 0.1603 0.2213 0.1519 0.2153 0.0146
Sample no. 31
Input data: 1.0000 0.9500
Target: 10.7680 2.0670 15.6720 8.4520 1.0620 8.3670
Output: 10.7627 2.0692 15.6822 8.4672 1.0631 8.3613
err(%): -0.0490 0.1071 0.0649 0.1804 0.1077 -0.0679
Sample no. 32
Input data: 1.0000 0.7000
Target: 14.4170 2.6180 21.1760 12.4860 1.0620 8.3490
Output: 14.3857 2.6196 21.1763 12.4954 1.0640 8.3499
err(%): -0.2169 0.0609 0.0016 0.0756 0.1924 0.0105
Sample no. 33
Input data: 1.0000 0.6500
Target: 15.6120 2.7770 22.7720 13.8080 1.0620 8.3430
Output: 15.6112 2.7686 22.7132 13.7832 1.0626 8.3395
err(%): -0.0052 -0.3034 -0.2581 -0.1797 0.0554 -0.0422
Sample no. 34
Input data: 0.8500 0.9000
Target: 8.8670 1.9050 14.0520 7.9240 1.0620 7.1090
Output: 8.8739 1.9038 14.0327 7.9187 1.0610 7.1097
err(%): 0.0782 -0.0633 -0.1371 -0.0665 -0.0903 0.0095
Sample no. 35
Input data: 0.8500 0.7500
Target: 10.4260 2.1820 16.8200 9.9460 1.0620 7.1000
Output: 10.4406 2.1804 16.8085 9.9424 1.0628 7.1055
err(%): 0.1405 -0.0714 -0.0682 -0.0357 0.0717 0.0773
Sample no. 36
Input data: 0.8500 0.6000
Target: 13.0840 2.5930 20.9330 13.3950 1.0620 7.0850
Output: 13.0940 2.5969 20.9758 13.4237 1.0610 7.0851
err(%): 0.0768 0.1499 0.2045 0.2139 -0.0903 0.0019
Sample no. 37
Input data: 0.7000 0.9000
Target: 6.7390 1.6570 11.5720 6.8190 1.0620 5.8540
Output: 6.7383 1.6560 11.5820 6.8410 1.0625 5.8541
err(%): -0.0099 -0.0631 0.0860 0.3221 0.0508 0.0025
Sample no. 38
Input data: 0.7000 0.8000
Target: 7.3970 1.8000 12.9980 7.8510 1.0620 5.8500
Output: 7.3851 1.7999 13.0090 7.8595 1.0609 5.8449
err(%): -0.1613 -0.0063 0.0848 0.1078 -0.1056 -0.0872
Sample no. 39
Input data: 0.7000 0.7000
Target: 8.3070 1.9820 14.8240 9.2790 1.0620 5.8430
Output: 8.2897 1.9858 14.8437 9.2882 1.0627 5.8380
err(%): -0.2078 0.1908 0.1329 0.0996 0.0699 -0.0850
Sample no. 40
Input data: 0.7000 0.6000
Target: 9.6370 2.2240 17.2390 11.3660 1.0620 5.8340
Output: 9.6226 2.2265 17.2506 11.3855 1.0617 5.8309
err(%): -0.1494 0.1107 0.0671 0.1719 -0.0273 -0.0527
Sample no. 41
Input data: 1.0000 0.6000
Target: 17.0810 2.9620 24.6250 15.4310 1.0620 8.3360
Output: 17.1326 2.9648 24.6529 15.4436 1.0601 8.3359
err(%): 0.3021 0.0949 0.1133 0.0818 -0.1745 -0.0009
Sample no. 42
Input data: 0.9500 0.7000
Target: 13.2790 2.5120 20.1180 11.9510 1.0620 7.9320
Output: 13.2658 2.5173 20.1531 11.9653 1.0615 7.9306
err(%): -0.0991 0.2114 0.1743 0.1200 -0.0428 -0.0177
Sample no. 43
Input data: 0.9500 0.6500
Target: 14.3590 2.6630 21.6340 13.2070 1.0620 7.9260
Output: 14.3434 2.6639 21.6401 13.2152 1.0632 7.9216
err(%): -0.1088 0.0334 0.0280 0.0622 0.1121 -0.0549
Sample no. 44
Input data: 0.9500 0.6000
Target: 15.6870 2.8390 23.3940 14.7520 1.0620 7.9190
Output: 15.7137 2.8381 23.3760 14.7367 1.0627 7.9167
err(%): 0.1704 -0.0308 -0.0769 -0.1040 0.0641 -0.0296
Sample no. 45
Input data: 0.9500 0.5500
Target: 17.3500 3.0460 25.4620 16.6860 1.0620 7.9110
Output: 17.3649 3.0615 25.5968 16.7768 1.0628 7.9100
err(%): 0.0860 0.5085 0.5296 0.5443 0.0772 -0.0131
Sample no. 46
Input data: 0.9500 0.5000
Target: 18.5740 3.1220 26.2200 18.1030 1.0570 7.9040
Output: 18.7269 3.1439 26.4130 18.2210 1.0558 7.8998
err(%): 0.8232 0.7014 0.7359 0.6516 -0.1160 -0.0533
Sample no. 47
Input data: 0.9000 0.7000
Target: 12.1880 2.4060 19.0590 11.4150 1.0620 7.5140
Output: 12.1978 2.4034 19.0453 11.3950 1.0608 7.5189
err(%): 0.0802 -0.1067 -0.0718 -0.1751 -0.1146 0.0657
Sample no. 48
Input data: 0.9000 0.6500
Target: 13.1600 2.5500 20.4960 12.6080 1.0620 7.5090
Output: 13.1556 2.5473 20.4786 12.5927 1.0613 7.5096
err(%): -0.0337 -0.1074 -0.0850 -0.1211 -0.0657 0.0086
Sample no. 49
Input data: 0.9000 0.6000
Target: 14.3550 2.7160 22.1640 14.0730 1.0620 7.5020
Output: 14.3617 2.7132 22.1452 14.0606 1.0632 7.5004
err(%): 0.0470 -0.1046 -0.0850 -0.0880 0.1168 -0.0210
Sample no. 50
Input data: 0.9000 0.5500
Target: 15.8500 2.9120 24.1230 15.9060 1.0620 7.4940
Output: 15.8826 2.9158 24.1437 15.9093 1.0638 7.4979
err(%): 0.2057 0.1305 0.0856 0.0208 0.1679 0.0519

The prediction data sets of the Bfp simulation rig and the prediction outputs of the trained neural network with their prediction errors in percentages are listed as follows. (The errors are plotted in Figure 5.9 to Figure 5.14.)
Sample no. 1:
Input data: 1.0000 0.8500
Target: 11.8960 2.2500 17.4920 9.6990 1.0620 8.3610
Output: 11.8974 2.2466 17.5048 9.6998 1.0584 8.3536
err(%): -0.0119 0.1497 -0.0732 -0.0087 0.3385 0.0890
Sample no. 2:
Input data: 1.0000 0.8000
Target: 12.5990 2.3570 18.5700 10.4770 1.0620 8.3580
Output: 12.6141 2.3585 18.6061 10.4953 1.0586 8.3518
err(%): -0.1196 -0.0623 -0.1946 -0.1746 0.3237 0.0743
Sample no. 3:
Input data: 0.9500 0.8000
Target: 11.6340 2.2640 17.6420 10.0380 1.0620 7.9400
Output: 11.6425 2.2611 17.6387 10.0327 1.0599 7.9377
err(%): -0.0728 0.1270 0.0189 0.0524 0.1991 0.0294
Sample no. 4:
Input data: 0.9500 0.7500
Target: 12.3840 2.3800 18.7990 10.9100 1.0620 7.9360
Output: 12.3927 2.3803 18.8139 10.9143 1.0599 7.9343
err(%): -0.0704 -0.0137 -0.0792 -0.0394 0.1945 0.0211
Sample no. 5:
Input data: 0.9000 0.8000
Target: 10.7080 2.1710 16.7130 9.6000 1.0620 7.5220
Output: 10.7204 2.1646 16.6581 9.5686 1.0619 7.5262
err(%): -0.1158 0.2940 0.3286 0.3273 0.0122 -0.0561
Sample no. 6:
Input data: 0.8000 0.8500
Target: 8.5170 1.8990 13.9940 8.0960 1.0620 6.6890
Output: 8.5070 1.8935 13.9375 8.0710 1.0606 6.6840
err(%): 0.1172 0.2907 0.4035 0.3087 0.1311 0.0752
Sample no. 7:
Input data: 0.8000 0.9500
Target: 7.7850 1.7540 12.5370 7.0870 1.0620 6.6930
Output: 7.7679 1.7499 12.5136 7.0769 1.0602 6.6922

err(%): 0.2196 0.2364 0.1866 0.1420 0.1699 0.0119
Sample no. 8:
Input data: 0.7000 0.7500
Target: 7.8120 1.8850 13.8510 8.5030 1.0620 5.8470
Output: 7.7953 1.8827 13.8307 8.4887 1.0613 5.8422
err(%): 0.2132 0.1220 0.1469 0.1679 0.0622 0.0827
Sample no. 9:
Input data: 0.7000 0.6500
Target: 8.9040 2.0940 15.9410 10.2160 1.0620 5.8390
Output: 8.8885 2.0931 15.9270 10.2134 1.0631 5.8361
err(%): 0.1742 0.0424 0.0880 0.0253 -0.1069 0.0495
Sample no. 10:
Input data: 0.6000 0.7000
Target: 6.6530 1.7700 12.7050 8.2140 1.0620 5.0080
Output: 6.6639 1.7687 12.6847 8.1870 1.0612 5.0134
err(%): -0.1635 0.0716 0.1595 0.3286 0.0752 -0.1078
Sample no. 11:
Input data: 0.8000 0.7500
Target: 9.5110 2.0830 15.8310 9.4650 1.0620 6.6830
Output: 9.5092 2.0773 15.7671 9.4367 1.0626 6.6818
err(%): 0.0193 0.2726 0.4034 0.2991 -0.0577 0.0181
Sample no. 12:
Input data: 0.8000 0.7000
Target: 10.1510 2.1940 16.9420 10.3460 1.0620 6.6790
Output: 10.1549 2.1886 16.8892 10.3218 1.0628 6.6814
err(%): -0.0381 0.2480 0.3115 0.2338 -0.0785 -0.0359
Sample no. 13:
Input data: 0.8000 0.6000
Target: 11.8740 2.4700 19.7020 12.7180 1.0620 6.6680
Output: 11.8772 2.4664 19.6645 12.7065 1.0597 6.6693
err(%): -0.0272 0.1440 0.1906 0.0903 0.2179 -0.0192
Sample no. 14:
Input data: 0.7500 0.6000
Target: 10.7250 2.3470 18.4710 12.0420 1.0620 6.2510
Output: 10.7152 2.3435 18.4265 12.0330 1.0601 6.2496
err(%): 0.0912 0.1484 0.2410 0.0751 0.1801 0.0223
Sample no. 15:
Input data: 0.7500 0.7000
Target: 9.2050 2.0880 15.8830 9.8120 1.0620 6.2610
Output: 9.1921 2.0852 15.8453 9.7992 1.0628 6.2576
err(%): 0.1397 0.1334 0.2372 0.1301 -0.0786 0.0539
Sample no. 16:
Input data: 0.7500 0.8000
Target: 8.1650 1.8930 13.9270 8.2870 1.0620 6.2680
Output: 8.1464 1.8884 13.8880 8.2708 1.0607 6.2608
err(%): 0.2283 0.2426 0.2803 0.1951 0.1230 0.1144
Sample no. 17:
Input data: 0.7500 0.9000
Target: 7.4160 1.7400 12.3990 7.1870 1.0620 6.2730
Output: 7.3987 1.7351 12.3745 7.1864 1.0608 6.2702
err(%): 0.2336 0.2827 0.1974 0.0082 0.1117 0.0447
Sample no. 18:
Input data: 0.6500 0.8000
Target: 6.6670 1.7070 12.0690 7.4130 1.0620 5.4320
Output: 6.6760 1.7023 12.0436 7.3997 1.0612 5.4350
err(%): -0.1356 0.2745 0.2105 0.1795 0.0738 -0.0549
Sample no. 19:
Input data: 0.6500 0.7000
Target: 7.4560 1.8760 13.7640 8.7460 1.0620 5.4260
Output: 7.4499 1.8754 13.7547 8.7289 1.0622 5.4261
err(%): 0.0813 0.0330 0.0675 0.1957 -0.0162 -0.0010
Sample no. 20:
Input data: 0.6500 0.6000
Target: 8.6090 2.1010 16.0080 10.6920 1.0620 5.4170
Output: 8.6004 2.0999 16.0014 10.6893 1.0632 5.4160
err(%): 0.1004 0.0527 0.0415 0.0256 -0.1132 0.0180

Appendix B Training and prediction results of the Test transmission rig

Training data sets of the test transmissin rig and the corresponding outputs of the trained neural network with the output errors in percentages are listed as follows.(The errors are plotted in Figure 5.16 to Figure 5.21.)

Sample no. 1
Input data: 2.6800 2.5800
Target: 0.8300 0.6500 1.1000 1.4800 0.2400 2.6000
Output: 0.8752 0.6592 1.1370 1.4823 0.2436 2.5124
err(%): 5.4476 1.4164 3.3599 0.1553 1.5049 -3.3674
Sample no. 2
Input data: 2.8800 2.5800
Target: 0.9200 0.6900 1.2000 1.5400 0.2600 2.8500
Output: 0.9848 0.7016 1.2237 1.5602 0.2452 2.7785
err(%): 7.0414 1.6834 1.9757 1.3104 -5.6761 -2.5073
Sample no. 3
Input data: 3.0900 2.5800
Target: 1.1300 0.7400 1.3000 1.6200 0.2500 3.0600
Output: 1.1031 0.7449 1.3172 1.6340 0.2546 3.0550
err(%): -2.3819 0.6570 1.3264 0.8616 1.8337 -0.1634
Sample no. 4
Input data: 3.2900 2.5800
Target: 1.2600 0.7800 1.4000 1.6900 0.2600 3.3200
Output: 1.2340 0.7854 1.4063 1.7036 0.2572 3.2936
err(%): -2.0604 0.6876 0.4482 0.8049 -1.0916 -0.7954
Sample no. 5
Input data: 3.4900 2.5800
Target: 1.2800 0.8300 1.4900 1.7800 0.2600 3.5400
Output: 1.3707 0.8242 1.4975 1.7723 0.2628 3.5310
err(%): 7.0853 -0.6965 0.5044 -0.4339 1.0623 -0.2542
Sample no. 6
Input data: 2.6700 2.7900
Target: 0.7800 0.6400 1.0800 1.4400 0.2500 2.6000
Output: 0.7443 0.6200 1.0094 1.3398 0.2497 2.6058
err(%): -4.5717 -3.1248 -6.5360 -6.9568 -0.1013 0.2226
Sample no. 7
Input data: 2.8800 2.7900
Target: 0.8500 0.6900 1.1700 1.5200 0.2600 2.8600
Output: 0.8585 0.6678 1.1389 1.4525 0.2521 2.8620
err(%): 0.9944 -3.2143 -2.6542 -4.4420 -3.0518 0.0682
Sample no. 8
Input data: 3.0900 2.7900
Target: 1.0600 0.7300 1.2800 1.5900 0.2500 3.0700
Output: 0.9768 0.7190 1.2557 1.5574 0.2579 3.1182
err(%): -7.8537 -1.5010 -1.8952 -2.0529 3.1558 1.5690
Sample no. 9
Input data: 3.2900 2.7900
Target: 1.1600 0.7900 1.3600 1.6700 0.2500 3.3300
Output: 1.1115 0.7631 1.3634 1.6439 0.2570 3.3446
err(%): -4.1774 -3.4060 0.2513 -1.5640 2.7956 0.4398
Sample no. 10
Input data: 3.4800 2.7900
Target: 1.2000 0.8200 1.4700 1.7400 0.2500 3.5700
Output: 1.2413 0.8118 1.4518 1.7227 0.2559 3.5625
err(%): 3.4428 -0.9940 -1.2370 -0.9919 2.3700 -0.2104
Sample no. 11
Input data: 2.6900 2.9900
Target: 0.7200 0.6200 1.0100 1.3000 0.2500 2.5800
Output: 0.6891 0.6265 1.0140 1.3515 0.2505 2.6395
err(%): -4.2894 1.0547 0.4005 3.9609 0.2193 2.3073
Sample no. 12
Input data: 2.8800 2.9900
Target: 0.7400 0.6500 1.0900 1.3700 0.2400 2.8600
Output: 0.7836 0.6603 1.0992 1.3954 0.2496 2.8487
err(%): 5.8874 1.5836 0.8444 1.8538 4.0069 -0.3957
Sample no. 13
Input data: 3.0800 2.9900
Target: 0.9100 0.7000 1.2000 1.4500 0.2600 3.0600
Output: 0.8827 0.6968 1.1874 1.4496 0.2455 3.0952
err(%): -3.0050 -0.4523 -1.0496 -0.0272 -5.5691 1.1503
Sample no. 14
Input data: 3.2900 2.9900
Target: 1.1000 0.7400 1.2900 1.5100 0.2600 3.3200
Output: 1.0060 0.7386 1.2890 1.5169 0.2519 3.3404
err(%): -8.5422 -0.1914 -0.0790 0.4601 -3.1011 0.6158
Sample no. 15
Input data: 3.4800 2.9900
Target: 1.1300 0.7800 1.3800 1.5600 0.2400 3.5600
Output: 1.1387 0.7758 1.3774 1.5749 0.2565 3.5648
err(%): 0.7693 -0.5408 -0.1889 0.9523 6.8943 0.1335
Sample no. 16
Input data: 3.6900 2.9900
Target: 1.2100 0.8000 1.4700 1.6100 0.2600 3.7600
Output: 1.2709 0.8185 1.4716 1.6353 0.2505 3.8060
err(%): 5.0316 2.3119 0.1095 1.5744 -3.6432 1.2236
Sample no. 17
Input data: 2.6800 3.1900
Target: 0.6800 0.6000 0.9600 1.2400 0.2400 2.6100
Output: 0.6215 0.5915 0.9507 1.2168 0.2552 2.5831
err(%): -8.5995 -1.4209 -0.9686 -1.8722 6.3474 -1.0320
Sample no. 18
Input data: 2.8700 3.1900
Target: 0.7800 0.6600 1.0300 1.3000 0.2600 2.8500
Output: 0.7161 0.6270 1.0382 1.2807 0.2481 2.8256
err(%): -8.1977 -5.0001 0.7932 -1.4834 -4.5624 -0.8579
Sample no. 19
Input data: 3.0800 3.1900
Target: 0.8800 0.6800 1.1200 1.3500 0.2500 3.0700
Output: 0.8328 0.6785 1.1287 1.3507 0.2520 3.1005
err(%): -5.3679 -0.2205 0.7804 0.0551 0.7957 0.9942
Sample no. 20
Input data: 3.2900 3.1900
Target: 0.9900 0.7200 1.2100 1.4200 0.2500 3.3200
Output: 0.9542 0.7195 1.2177 1.4127 0.2514 3.3523
err(%): -3.6194 -0.0647 0.6324 -0.5111 0.5506 0.9721
Sample no. 21
Input data: 3.4800 3.1900
Target: 1.0400 0.7500 1.3000 1.4600 0.2700 3.5600
Output: 1.0691 0.7556 1.2971 1.4713 0.2508 3.5722
err(%): 2.7976 0.7494 -0.2258 0.7739 -7.0931 0.3430
Sample no. 22
Input data: 3.6900 3.1900
Target: 1.2000 0.8000 1.3900 1.5300 0.2400 3.7900
Output: 1.1902 0.7935 1.3877 1.5286 0.2597 3.8139
err(%): -0.8208 -0.8099 -0.1673 -0.0903 8.2182 0.6297
Sample no. 23
Input data: 3.8900 3.1900
Target: 1.4000 0.8500 1.4800 1.5800 0.2400 3.9000
Output: 1.3145 0.8329 1.4722 1.5868 0.2519 4.0316
err(%): -6.1072 -2.0110 -0.5249 0.4314 4.9469 3.3747
Sample no. 24
Input data: 2.6800 3.4000
Target: 0.6000 0.5500 0.9100 1.1300 0.2400 2.6300
Output: 0.6143 0.5909 0.8992 1.1561 0.2509 2.5407
err(%): 2.3915 7.4408 -1.1822 2.3104 4.5418 -3.3948
Sample no. 25
Input data: 2.8800 3.4000
Target: 0.6700 0.6100 0.9800 1.1700 0.2500 2.8300
Output: 0.6988 0.6107 0.9877 1.2009 0.2428 2.8105
err(%): 4.2971 0.1159 0.7894 2.6376 -2.8605 -0.6906
Sample no. 26
Input data: 3.0900 3.4000
Target: 0.7300 0.6300 1.0700 1.2400 0.2500 3.0700
Output: 0.7893 0.6475 1.0724 1.2448 0.2438 3.0769
err(%): 8.1292 2.7754 0.2224 0.3884 -2.4895 0.2245
Sample no. 27
Input data: 3.2900 3.4000
Target: 0.8200 0.6800 1.1500 1.3000 0.2500 3.3300
Output: 0.8765 0.6776 1.1549 1.2965 0.2461 3.3183
err(%): 6.8906 -0.3535 0.4264 -0.2727 -1.5432 -0.3516
Sample no. 28
Input data: 3.4800 3.4000
Target: 0.9400 0.6900 1.2300 1.3200 0.2600 3.5400
Output: 0.9670 0.7141 1.2321 1.3492 0.2476 3.5534
err(%): 2.8757 3.4870 0.1674 2.2158 -4.7637 0.3780

Sample no. 29
Input data: 3.6900 3.4000
Target: 1.1300 0.7400 1.3100 1.3900 0.2700 3.7800
Output: 1.0807 0.7433 1.3175 1.3929 0.2527 3.7990
err(%): -4.3634 0.4406 0.5717 0.2060 -6.4253 0.5020
Sample no. 30
Input data: 3.9000 3.3900
Target: 1.2900 0.7700 1.4000 1.4400 0.2500 4.0300
Output: 1.2184 0.7799 1.4021 1.4495 0.2609 4.0364
err(%): -5.5542 1.2918 0.1528 0.6609 4.3784 0.1591
Sample no. 31
Input data: 4.0900 3.4000
Target: 1.3700 0.8000 1.4800 1.4800 0.2600 4.2800
Output: 1.3452 0.8077 1.4733 1.4899 0.2563 4.2502
err(%): -1.8093 0.9566 -0.4504 0.6712 -1.4323 -0.6960
Sample no. 32
Input data: 2.6900 3.6000
Target: 0.5400 0.5400 0.8600 1.0500 0.2600 2.6200
Output: 0.5842 0.5397 0.8649 1.0598 0.2638 2.6047
err(%): 8.1919 -0.0648 0.5687 0.9304 1.4564 -0.5846
Sample no. 33
Input data: 2.8900 3.6000
Target: 0.6200 0.5900 0.9300 1.1000 0.2600 2.8700
Output: 0.6504 0.5729 0.9390 1.1056 0.2608 2.8457
err(%): 4.9095 -2.8975 0.9682 0.5127 0.3077 -0.8451
Sample no. 34
Input data: 3.0900 3.6000
Target: 0.7000 0.6000 1.0100 1.1400 0.2600 3.1400
Output: 0.7279 0.6146 1.0131 1.1526 0.2592 3.0979
err(%): 3.9841 2.4274 0.3072 1.1010 -0.3111 -1.3392
Sample no. 35
Input data: 3.3000 3.6000
Target: 0.7800 0.6500 1.0900 1.2000 0.2600 3.3500
Output: 0.8223 0.6466 1.0963 1.1994 0.2582 3.3745
err(%): 5.4211 -0.5267 0.5818 -0.0474 -0.6917 0.7301
Sample no. 36
Input data: 3.2700 3.6000
Target: 0.8000 0.6300 1.0800 1.1800 0.2500 3.3100
Output: 0.7971 0.6430 1.0800 1.1922 0.2595 3.3253
err(%): -0.3664 2.0583 0.0005 1.0354 3.7849 0.4623
Sample no. 37
Input data: 3.6700 3.6000
Target: 0.9600 0.7000 1.2300 1.2800 0.2500 3.7400
Output: 1.0047 0.7073 1.2400 1.2839 0.2533 3.8088
err(%): 4.6609 1.0409 0.8138 0.3054 1.3255 1.8400
Sample no. 38
Input data: 3.8800 3.6000
Target: 1.1200 0.7400 1.3200 1.3200 0.2500 4.0800
Output: 1.1114 0.7401 1.3151 1.3316 0.2506 4.0233
err(%): -0.7685 0.0139 -0.3715 0.8803 0.2582 -1.3902
Sample no. 39
Input data: 4.0900 3.6000
Target: 1.2500 0.7700 1.4000 1.3700 0.2600 4.2700
Output: 1.2405 0.7773 1.3979 1.3760 0.2489 4.2873
err(%): -0.7583 0.9513 -0.1531 0.4408 -4.2875 0.4056
Sample no. 40
Input data: 4.2900 3.6000
Target: 1.3800 0.8000 1.4800 1.4200 0.2500 4.5000
Output: 1.3630 0.8077 1.4724 1.4197 0.2537 4.4997
err(%): -1.2339 0.9674 -0.5115 -0.0194 1.4857 -0.0062
Sample no. 41
Input data: 4.4900 3.6000
Target: 1.5200 0.8400 1.5600 1.4800 0.2600 4.7600
Output: 1.4882 0.8364 1.5482 1.4671 0.2523 4.7056
err(%): -2.0899 -0.4297 -0.7586 -0.8719 -2.9510 -1.1438
Sample no. 42
Input data: 2.8900 3.8000
Target: 0.6100 0.5600 0.8800 1.0300 0.2600 2.8700
Output: 0.6067 0.5523 0.9042 1.0426 0.2623 2.8728
err(%): -0.5430 -1.3799 2.7457 1.2190 0.8771 0.0962
Sample no. 43
Input data: 3.0900 3.8000
Target: 0.6800 0.5900 0.9600 1.0800 0.2500 3.0700
Output: 0.6854 0.5879 0.9674 1.0826 0.2605 3.1142
err(%): 0.7889 -0.3554 0.7750 0.2427 4.1918 1.4392
Sample no. 44
Input data: 3.3000 3.8000
Target: 0.7500 0.6100 1.0300 1.1000 0.2500 3.4000
Output: 0.7722 0.6229 1.0416 1.1279 0.2551 3.3510
err(%): 2.9577 2.1078 1.1289 2.5356 2.0316 -1.4410
Sample no. 45
Input data: 2.6600 3.8000
Target: 0.5200 0.5100 0.8100 0.9700 0.2600 2.6000
Output: 0.5251 0.5186 0.7999 0.9713 0.2546 2.6043
err(%): 0.9877 1.6770 -1.2422 0.1364 -2.0799 0.1655
Sample no. 46
Input data: 3.4700 3.8000
Target: 0.8200 0.6400 1.1000 1.1600 0.2500 3.6200
Output: 0.8502 0.6433 1.1060 1.1532 0.2530 3.5783
err(%): 3.6822 0.5101 0.5459 -0.5836 1.2051 -1.1530
Sample no. 47
Input data: 3.6700 3.8000
Target: 0.9200 0.6700 1.1800 1.1900 0.2500 3.7600
Output: 0.9482 0.6762 1.1813 1.2017 0.2508 3.8407
err(%): 3.0704 0.9262 0.1122 0.9792 0.3214 2.1452
Sample no. 48
Input data: 3.8800 3.8000
Target: 1.0500 0.7000 1.2500 1.2400 0.2600 4.0100
Output: 1.0510 0.7080 1.2577 1.2427 0.2495 4.0556
err(%): 0.0941 1.1444 0.6143 0.2144 -4.0357 1.1362
Sample no. 49
Input data: 4.0800 3.8000
Target: 1.1500 0.7400 1.3200 1.3000 0.2500 4.2400
Output: 1.1605 0.7372 1.3259 1.2839 0.2537 4.2633
err(%): 0.9150 -0.3810 0.4500 -1.2413 1.4703 0.5484
Sample no. 50
Input data: 4.2900 3.8000
Target: 1.2300 0.7900 1.4100 1.3400 0.2500 4.5100
Output: 1.2795 0.7726 1.3961 1.3358 0.2514 4.4836
err(%): 4.0279 -2.2017 -0.9829 -0.3125 0.5769 -0.5845
Sample no. 51
Input data: 4.5000 3.8000
Target: 1.3900 0.8000 1.4700 1.3700 0.2500 4.8100
Output: 1.3967 0.8153 1.4741 1.3836 0.2495 4.7141
err(%): 0.4847 1.9065 0.2766 0.9931 -0.1964 -1.9933
Sample no. 52
Input data: 4.7000 3.8000
Target: 1.5400 0.8400 1.5500 1.4300 0.2500 5.0100
Output: 1.5205 0.8425 1.5405 1.4211 0.2491 4.9589
err(%): -1.2684 0.2986 -0.6112 -0.6225 -0.3490 -1.0195
Sample no. 53
Input data: 2.6800 4.0000
Target: 0.5200 0.5100 0.7800 0.9500 0.2600 2.6100
Output: 0.4974 0.5035 0.7890 0.9460 0.2580 2.6580
err(%): -4.3425 -1.2651 1.1485 -0.4227 -0.7632 1.8386
Sample no. 54
Input data: 2.8800 4.0000
Target: 0.5400 0.5500 0.8500 0.9900 0.2500 2.8400
Output: 0.5625 0.5351 0.8519 0.9894 0.2581 2.8702
err(%): 4.1602 -2.7152 0.2205 -0.0573 3.2454 1.0621
Sample no. 55
Input data: 3.0800 4.0000
Target: 0.6600 0.5700 0.9200 1.0300 0.2600 3.0600
Output: 0.6267 0.5727 0.9193 1.0317 0.2543 3.0907
err(%): -5.0410 0.4786 -0.0798 0.1610 -2.2014 1.0028
Sample no. 56
Input data: 3.2700 4.0000
Target: 0.6800 0.6000 0.9800 1.0400 0.2500 3.3200
Output: 0.7072 0.6006 0.9875 1.0702 0.2563 3.3075
err(%): 4.0029 0.1011 0.7607 2.8997 2.5260 -0.3759
Sample no. 57
Input data: 3.4700 4.0000
Target: 0.7800 0.6100 1.0500 1.0800 0.2600 3.5000
Output: 0.7910 0.6332 1.0568 1.0991 0.2525 3.5548
err(%): 1.4143 3.8092 0.6454 1.7722 -2.8841 1.5643
Sample no. 58
Input data: 3.6800 4.0000
Target: 0.8500 0.6500 1.1200 1.1300 0.2500 3.6500
Output: 0.8857 0.6562 1.1293 1.1330 0.2551 3.7851
err(%): 4.1959 0.9571 0.8320 0.2646 2.0316 3.7004
Sample no. 59
Input data: 3.8800 4.0000

Target: 0.9500 0.6900 1.1900 1.1600 0.2600 4.0100
Output: 0.9711 0.6830 1.1936 1.1706 0.2526 3.9578
err(%): 2.2205 -1.0192 0.3035 0.9111 -2.8567 -1.3028
Sample no. 60
Input data: 4.0800 4.0000
Target: 1.0800 0.7300 1.2600 1.2200 0.2500 4.2400
Output: 1.0774 0.7195 1.2627 1.2065 0.2545 4.2107
err(%): -0.2420 -1.4422 0.2145 -1.1071 1.7838 -0.6901
Sample no. 61
Input data: 4.2900 4.0000
Target: 1.2000 0.7600 1.3300 1.2600 0.2600 4.5200
Output: 1.1982 0.7591 1.3335 1.2538 0.2511 4.4668
err(%): -0.1462 -0.1221 0.2629 -0.4882 -3.4172 -1.1761
Sample no. 62
Input data: 4.4900 4.0000
Target: 1.2900 0.7900 1.4100 1.2800 0.2400 4.7700
Output: 1.3184 0.7931 1.3982 1.2966 0.2540 4.7124
err(%): 2.1985 0.3906 -0.8394 1.2956 5.8343 -1.2077
Sample no. 63
Input data: 4.7000 3.9900
Target: 1.4000 0.8400 1.4800 1.3300 0.2600 5.0000
Output: 1.4434 0.8266 1.4736 1.3329 0.2458 4.9570
err(%): 3.0989 -1.5980 -0.4341 0.2180 -5.4457 -0.8608
Sample no. 64
Input data: 4.9000 4.0000
Target: 1.5000 0.8400 1.5300 1.3500 0.2600 5.1900
Output: 1.5517 0.8628 1.5357 1.3665 0.2517 5.1715
err(%): 3.4495 2.7202 0.3742 1.2207 -3.1892 -0.3574
Sample no. 65
Input data: 2.8800 4.2100
Target: 0.5300 0.5500 0.8200 0.9800 0.2500 2.9000
Output: 0.5120 0.5269 0.8167 0.9234 0.2626 2.8624
err(%): -3.4005 -4.1941 -0.3976 -5.7724 5.0399 -1.2955
Sample no. 66
Input data: 3.0800 4.2000
Target: 0.6000 0.5800 0.8900 1.0100 0.2400 3.0600
Output: 0.5840 0.5665 0.8862 0.9880 0.2576 3.1179
err(%): -2.6684 -2.3353 -0.4254 -2.1753 7.3244 1.8916
Sample no. 67
Input data: 3.2900 4.2000
Target: 0.7200 0.5800 0.9600 1.0400 0.2700 3.3500
Output: 0.6622 0.6039 0.9589 1.0398 0.2495 3.3468
err(%): -8.0314 4.1251 -0.1134 -0.0214 -7.5773 -0.0946
Sample no. 68
Input data: 3.6700 4.2000
Target: 0.8400 0.6700 1.0800 1.1100 0.2600 3.8800
Output: 0.8396 0.6507 1.0943 1.1151 0.2584 3.8127
err(%): -0.0496 -2.8794 1.3219 0.4603 -0.6231 -1.7355
Sample no. 69
Input data: 3.8800 4.2000
Target: 0.9700 0.7000 1.1500 1.1600 0.2600 4.0300
Output: 0.9466 0.6924 1.1619 1.1539 0.2590 4.0996
err(%): -2.4134 -1.0838 1.0360 -0.5282 -0.3869 1.7260
Sample no. 70
Input data: 4.0800 4.2000
Target: 1.0300 0.7200 1.2100 1.1800 0.2700 4.2900
Output: 1.0492 0.7265 1.2221 1.1927 0.2591 4.3087
err(%): 1.8672 0.9050 1.0022 1.0728 -4.0555 0.4366
Sample no. 71
Input data: 4.2900 4.2000
Target: 1.1800 0.7600 1.2800 1.2200 0.2500 4.5300
Output: 1.1571 0.7569 1.2837 1.2252 0.2633 4.5373
err(%): -1.9378 -0.4013 0.2877 0.4259 5.3283 0.1606
Sample no. 72
Input data: 4.6900 4.2000
Target: 1.3600 0.8200 1.4200 1.3000 0.2500 5.0300
Output: 1.3846 0.8196 1.4035 1.2911 0.2546 4.9666
err(%): 1.8107 -0.0428 -1.1650 -0.6880 1.8205 -1.2597
Sample no. 73
Input data: 5.0200 4.2000
Target: 1.5500 0.8700 1.4700 1.3700 0.2400 5.2100
Output: 1.5715 0.8692 1.5065 1.3518 0.2497 5.3176
err(%): 1.3885 -0.0874 2.4804 -1.3280 4.0402 2.0646
Sample no. 74
Input data: 2.6800 4.4000
Target: 0.5000 0.5000 0.7200 0.9000 0.2500 2.6200
Output: 0.4662 0.5008 0.7159 0.9009 0.2507 2.6150
err(%): -6.7557 0.1629 -0.5757 0.0980 0.2960 -0.1925
Sample no. 75
Input data: 2.8800 4.4000
Target: 0.5200 0.5300 0.7900 0.9300 0.2700 2.8500
Output: 0.5294 0.5297 0.7764 0.9383 0.2485 2.8382
err(%): 1.8157 -0.0533 -1.7160 0.8948 -7.9580 -0.4144
Sample no. 76
Input data: 3.0900 4.4000
Target: 0.5600 0.5500 0.8500 0.9600 0.2700 3.1300
Output: 0.5953 0.5628 0.8486 0.9745 0.2594 3.0848
err(%): 6.2949 2.3187 -0.1628 1.5117 -3.9333 -1.4454
Sample no. 77
Input data: 3.2900 4.4000
Target: 0.6500 0.5900 0.9200 1.0000 0.2700 3.3400
Output: 0.6620 0.5900 0.9151 1.0065 0.2661 3.3396
err(%): 1.8430 -0.0027 -0.5279 0.6490 -1.4492 -0.0120
Sample no. 78
Input data: 3.4800 4.4000
Target: 0.7400 0.6300 0.9800 1.0300 0.2600 3.5400
Output: 0.7331 0.6192 0.9805 1.0387 0.2693 3.5679
err(%): -0.9377 -1.7149 0.0518 0.8443 3.5589 0.7868
Sample no. 79
Input data: 3.8900 4.4000
Target: 0.9400 0.6700 1.1100 1.0900 0.2600 3.9800
Output: 0.9123 0.6857 1.1172 1.1096 0.2647 4.0467
err(%): -2.9490 2.3459 0.6525 1.8019 1.8155 1.6754
Sample no. 80
Input data: 4.0800 4.4000
Target: 0.9400 0.7100 1.1800 1.1400 0.2500 4.2400
Output: 1.0058 0.7061 1.1746 1.1329 0.2610 4.2385
err(%): 7.0037 -0.5553 -0.4551 -0.6243 4.4106 -0.0355
Sample no. 81
Input data: 4.3000 4.4000
Target: 1.0400 0.7400 1.2500 1.1800 0.2600 4.5400
Output: 1.1046 0.7408 1.2473 1.1751 0.2537 4.4854
err(%): 6.2161 0.1043 -0.2161 -0.4184 -2.4387 -1.2029
Sample no. 82
Input data: 4.4900 4.4000
Target: 1.1900 0.7800 1.3200 1.2900 0.2600 4.7800
Output: 1.1956 0.7705 1.3079 1.2118 0.2544 4.7191
err(%): 0.4682 -1.2174 -0.9171 -6.0600 -2.1488 -1.2736
Sample no. 83
Input data: 4.7000 4.4000
Target: 1.3300 0.8000 1.3900 1.2500 0.2400 5.0300
Output: 1.3120 0.8057 1.3772 1.2857 0.2558 4.9772
err(%): -1.3569 0.7145 -0.9177 2.8566 6.5828 -1.0490
Sample no. 84
Input data: 5.0200 4.4000
Target: 1.5200 0.8500 1.4500 1.3100 0.2400 5.2500
Output: 1.4970 0.8508 1.4704 1.3216 0.2450 5.3259
err(%): -1.5111 0.0986 1.4096 0.8884 2.0691 1.4461
Sample no. 85
Input data: 2.6800 4.6000
Target: 0.5100 0.4800 0.7200 0.8700 0.2500 2.6200
Output: 0.4569 0.4879 0.7256 0.8815 0.2480 2.6474
err(%): -10.4135 1.6387 0.7753 1.3245 -0.8019 1.0471
Sample no. 86
Input data: 2.8600 4.6000
Target: 0.5000 0.5200 0.7700 0.9200 0.2500 2.8400
Output: 0.5159 0.5100 0.7735 0.9067 0.2469 2.8322
err(%): 3.1711 -1.9224 0.4547 -1.4441 -1.2308 -0.2731
Sample no. 87
Input data: 3.0800 4.6000
Target: 0.5500 0.5400 0.8600 0.9500 0.2500 3.1000
Output: 0.5781 0.5496 0.8363 0.9516 0.2495 3.0827
err(%): 5.1126 1.7786 -2.7555 0.1701 -0.2047 -0.5595
Sample no. 88
Input data: 3.2900 4.6000
Target: 0.6200 0.5800 0.9100 0.9800 0.2500 3.3400
Output: 0.6435 0.5795 0.9130 0.9900 0.2516 3.3331
err(%): 3.7848 -0.0870 0.3315 1.0243 0.6225 -0.2060
Sample no. 89
Input data: 3.4700 4.6000
Target: 0.7000 0.6100 0.9700 1.0100 0.2600 3.5600
Output: 0.7047 0.6085 0.9692 1.0178 0.2519 3.5471

err(%): 0.6711 -0.2391 -0.0805 0.7751 -3.1247 -0.3617
Sample no. 90
Input data: 3.6800 4.6000
Target: 0.8600 0.6100 1.0300 1.0400 0.2600 3.7800
Output: 0.7883 0.6418 1.0373 1.0516 0.2567 3.8023
err(%): -8.3320 5.2074 0.7094 1.1190 -1.2875 0.5886
Sample no. 91
Input data: 3.8800 4.6000
Target: 0.8700 0.6600 1.1000 1.0900 0.2600 4.0000
Output: 0.8898 0.6547 1.0983 1.0801 0.2585 4.0334
err(%): 2.2741 -0.7958 -0.1561 -0.9046 -0.5802 0.8354
Sample no. 92
Input data: 4.0900 4.6000
Target: 1.0300 0.6900 1.1700 1.1200 0.2600 4.3200
Output: 0.9803 0.6860 1.1658 1.1210 0.2592 4.2614
err(%): -4.8265 -0.5861 -0.3631 0.0932 -0.2955 -1.3557
Sample no. 93
Input data: 4.2900 4.6000
Target: 1.0400 0.7300 1.2400 1.1700 0.2600 4.5500
Output: 1.0911 0.7169 1.2318 1.1557 0.2584 4.5227
err(%): 4.9151 -1.8010 -0.6647 -1.2256 -0.6296 -0.5994
Sample no. 94
Input data: 4.5000 4.6000
Target: 1.2500 0.7600 1.3100 1.2000 0.2600 4.7600
Output: 1.1899 0.7545 1.2997 1.1986 0.2575 4.7713
err(%): -4.8100 -0.7192 -0.7880 -0.1128 -0.9728 0.2384
Sample no. 95
Input data: 4.7000 4.6000
Target: 1.3100 0.8400 1.3700 1.2500 0.2500 5.0100
Output: 1.3046 0.7856 1.3614 1.2303 0.2572 4.9870
err(%): -0.4091 -6.4742 -0.6307 -1.5726 2.8852 -0.4601
Sample no. 96
Input data: 5.0200 4.6000
Target: 1.5200 0.8500 1.4400 1.3000 0.2600 5.2400
Output: 1.4786 0.8584 1.4472 1.2857 0.2505 5.3238
err(%): -2.7216 0.9863 0.4985 -1.0963 -3.6651 1.5994
Sample no. 97
Input data: 2.8700 4.8100
Target: 0.5300 0.5200 0.7900 0.9200 0.2500 2.8400
Output: 0.5440 0.5187 0.7918 0.9124 0.2544 2.8630
err(%): 2.6413 -0.2440 0.2289 -0.8292 1.7787 0.8085
Sample no. 98
Input data: 3.0600 4.8100
Target: 0.5400 0.5600 0.8400 0.9600 0.2500 3.0600
Output: 0.5920 0.5507 0.8437 0.9490 0.2543 3.0582
err(%): 9.6231 -1.6675 0.4446 -1.1447 1.7003 -0.0598
Sample no. 99
Input data: 3.2700 4.8100
Target: 0.6500 0.5900 0.9000 0.9900 0.2500 3.3000
Output: 0.6448 0.5918 0.9024 0.9909 0.2549 3.2924
err(%): -0.7948 0.3037 0.2622 0.0942 1.9704 -0.2304
Sample no. 100
Input data: 3.4700 4.8100
Target: 0.7700 0.6100 0.9700 1.0300 0.2600 3.5400
Output: 0.7166 0.6240 0.9609 1.0248 0.2543 3.5276
err(%): -6.9404 2.2967 -0.9344 -0.5045 -2.2031 -0.3495
Sample no. 101
Input data: 3.8800 4.8100
Target: 0.9200 0.7200 1.1000 1.1400 0.2500 4.1000
Output: 0.8934 0.6789 1.0935 1.0974 0.2600 4.0172
err(%): -2.8929 -5.7042 -0.5930 -3.7380 3.9829 -2.0190
Sample no. 102
Input data: 4.2900 4.8100
Target: 1.1100 0.7300 1.2400 1.1900 0.2500 4.5500
Output: 1.0877 0.7572 1.2263 1.1889 0.2551 4.5352
err(%): -2.0064 3.7262 -1.1074 -0.0894 2.0339 -0.3243
Sample no. 103
Input data: 4.7000 4.8100
Target: 1.2600 0.8000 1.3700 1.2500 0.2500 5.0100
Output: 1.2948 0.8031 1.3518 1.2576 0.2488 4.9982
err(%): 2.7652 0.3817 -1.3293 0.6061 -0.4864 -0.2346
Sample no. 104
Input data: 5.0200 4.8100
Target: 1.5200 0.8400 1.4300 1.3000 0.2500 5.2700
Output: 1.4541 0.8454 1.4421 1.3002 0.2455 5.3295
err(%): -4.3379 0.6408 0.8459 0.0120 -1.8120 1.1287

The prediction data sets of the test transmission rig and the neural network prediction outputs with their prediction errors in percentages are listed as follows.(The prediction errors are plotted in Figure 5.22 to Figure 5.27.)

Sample no. 1:
Input data: 4.4900 4.2000
Target: 1.2600 0.7900 1.3500 1.2600 0.2600 4.7600
Output: 1.2902 0.7998 1.3857 1.3127 0.2512 4.7890
err(%): -2.3973 -1.2444 -2.6472 -4.1853 3.4002 -0.6093
Sample no. 2:
Input data: 3.4700 4.2000
Target: 0.7900 0.6400 1.0100 1.0700 0.2500 3.5300
Output: 0.7746 0.6402 1.0419 1.1185 0.2571 3.5939
err(%): 1.9550 -0.0242 -3.1584 -4.5335 -2.8283 -1.8112
Sample no. 3:
Input data: 2.6700 4.2000
Target: 0.5000 0.5200 0.7500 0.9400 0.2500 2.6000
Output: 0.4935 0.5162 0.7744 0.9578 0.2585 2.6530
err(%): 1.2984 0.7362 -3.2490 -1.8980 -3.4084 -2.0367
Sample no. 4:
Input data: 3.0600 4.2100
Target: 0.6200 0.5700 0.8700 1.0100 0.2500 3.0800
Output: 0.6142 0.5756 0.8997 1.0352 0.2570 3.1009
err(%): 0.9375 -0.9909 -3.4140 -2.4944 -2.8155 -0.6799
Sample no. 5:
Input data: 3.4700 3.6000
Target: 0.8800 0.6700 1.1600 1.2500 0.2400 3.4300
Output: 0.8782 0.6795 1.1533 1.2578 0.2555 3.5909
err(%): 0.2093 -1.4135 0.5780 -0.6275 -6.4549 -4.6899
Sample no. 6:
Input data: 2.6900 3.8000
Target: 0.5000 0.5200 0.8200 0.9800 0.2600 2.6400
Output: 0.5159 0.5386 0.8176 1.0231 0.2603 2.6509
err(%): -3.1881 -3.5746 0.2954 -4.4010 -0.1252 -0.4140
Sample no. 7:
Input data: 3.6800 4.4000
Target: 0.8800 0.6500 1.0400 1.0700 0.2600 3.7800
Output: 0.8493 0.6638 1.0892 1.1288 0.2565 3.8424
err(%): 3.4867 -2.1222 -4.7332 -5.4942 1.3533 -1.6516
Sample no. 8:
Input data: 4.4900 4.8100
Target: 1.2400 0.7600 1.3000 1.2100 0.2500 4.7900
Output: 1.1933 0.7669 1.2988 1.2205 0.2538 4.7515
err(%): 3.7664 -0.9079 0.0929 -0.8654 -1.5077 0.8032
Sample no. 9:
Input data: 4.0900 4.8100
Target: 0.9600 0.7100 1.1700 1.1400 0.2600 4.3100
Output: 1.0065 0.7099 1.1788 1.1538 0.2555 4.3008
err(%): -4.8422 0.0135 -0.7517 -1.2121 1.7194 0.2142
Sample no. 10:
Input data: 3.2800 4.8100
Target: 0.6500 0.5800 0.9000 1.0000 0.2500 3.2700
Output: 0.6864 0.5844 0.9286 1.0149 0.2493 3.3596
err(%): -5.5924 -0.7628 -3.1833 -1.4865 0.2795 -2.7412
Sample no. 11:
Input data: 3.0800 4.8100
Target: 0.6000 0.5500 0.8400 0.9500 0.2500 3.0900
Output: 0.6218 0.5513 0.8701 0.9809 0.2471 3.1315
err(%): -3.6411 -0.2339 -3.5864 -3.2555 1.1594 -1.3444
Sample no. 12:
Input data: 2.6700 4.8100
Target: 0.4700 0.4900 0.7100 0.8600 0.2600 2.6400
Output: 0.5126 0.4844 0.7547 0.9082 0.2468 2.6820
err(%): -9.0680 1.1457 -6.2970 -5.6013 5.0728 -1.5919
Sample no. 13:
Input data: 3.2700 3.8000
Target: 0.7300 0.6100 1.0300 1.1100 0.2500 3.3100
Output: 0.7377 0.6303 1.0316 1.1559 0.2573 3.3478
err(%): -1.0564 -3.3356 -0.1575 -4.1312 -2.9274 -1.1428
Sample no. 14:
Input data: 3.2700 4.2000
Target: 0.6500 0.6100 0.9400 1.0500 0.2600 3.2800

Output: 0.6925 0.6089 0.9721 1.0785 0.2572 3.3517
err(%): -6.5443 0.1789 -3.4188 -2.7176 1.0936 -2.1853
Sample no. 15:
Input data: 2.8600 3.6000
Target: 0.6100 0.5600 0.9300 1.0900 0.2600 2.8000
Output: 0.5999 0.5767 0.9106 1.1080 0.2583 2.8385
err(%): 1.6622 -2.9834 2.0871 -1.6526 0.6558 -1.3739
Sample no. 16:
Input data: 2.8700 4.6000
Target: 0.5300 0.5100 0.7800 0.9200 0.2600 2.8500
Output: 0.5538 0.5276 0.8158 0.9582 0.2510 2.8923
err(%): -4.4872 -3.4508 -4.5854 -4.1476 3.4617 -1.4843
Sample no. 17:
Input data: 2.6600 4.0000
Target: 0.4500 0.4900 0.7700 0.9100 0.2600 2.6100
Output: 0.4949 0.5242 0.7862 0.9825 0.2602 2.6298
err(%): -9.9712 -6.9754 -2.1083 -7.9631 -0.0859 -0.7589
Sample no. 18:
Input data: 2.8600 4.0000
Target: 0.5200 0.5100 0.8400 0.9600 0.2600 2.8400
Output: 0.5566 0.5545 0.8533 1.0256 0.2588 2.8590
err(%): -7.0304 -8.7214 -1.5893 -6.8324 0.4509 -0.6686
Sample no. 19:
Input data: 3.0600 4.6000
Target: 0.5600 0.5500 0.8500 0.9600 0.2600 3.0700
Output: 0.6104 0.5582 0.8723 0.9925 0.2516 3.1071
err(%): -8.9919 -1.4835 -2.6288 -3.3849 3.2122 -1.2093
Sample no. 20:
Input data: 3.2700 4.6000
Target: 0.6300 0.5800 0.9100 0.9800 0.2600 3.2900
Output: 0.6806 0.5919 0.9368 1.0300 0.2531 3.3503
err(%): -8.0295 -2.0546 -2.9467 -5.1020 2.6483 -1.8318
Sample no. 21:
Input data: 3.0600 3.6000
Target: 0.7000 0.5800 1.0000 1.1400 0.2600 3.0800
Output: 0.6817 0.6095 0.9888 1.1574 0.2573 3.0821
err(%): 2.6202 -5.0843 1.1179 -1.5244 1.0531 -0.0680
Sample no. 22:
Input data: 3.2800 3.1900
Target: 0.9900 0.7100 1.2100 1.3900 0.2500 3.3100
Output: 0.9116 0.6891 1.1902 1.3614 0.2534 3.3291
err(%): 7.9190 2.9400 1.6369 2.0558 -1.3733 -0.5756
Sample no. 23:
Input data: 3.0600 4.0000
Target: 0.5900 0.5600 0.9000 1.0000 0.2500 3.0600
Output: 0.6270 0.5855 0.9230 1.0683 0.2581 3.0960
err(%): -6.2642 -4.5611 -2.5530 -6.8256 -3.2579 -1.1774
Sample no. 24:
Input data: 3.2900 4.0000
Target: 0.7000 0.6200 0.9900 1.0600 0.2500 3.3200
Output: 0.7184 0.6218 1.0057 1.1172 0.2576 3.3751
err(%): -2.6273 -0.2925 -1.5838 -5.3952 -3.0402 -1.6597
Sample no. 25:
Input data: 2.6700 3.4000
Target: 0.5900 0.5600 0.9000 1.1100 0.2400 2.5800
Output: 0.5644 0.5599 0.8749 1.1134 0.2578 2.5981
err(%): 4.3406 0.0255 2.7900 -0.3029 -7.4053 -0.7017
Sample no. 26:
Input data: 2.8700 3.4000
Target: 0.6300 0.5900 0.9800 1.1500 0.2500 2.8400
Output: 0.6436 0.5931 0.9556 1.1673 0.2565 2.8377
err(%): -2.1590 -0.5234 2.4878 -1.5051 -2.6100 0.0795
Sample no. 27:
Input data: 2.8600 4.4000
Target: 0.5000 0.5300 0.7800 0.9200 0.2700 2.8400
Output: 0.5472 0.5358 0.8216 0.9738 0.2548 2.8748
err(%): -9.4401 -1.0857 -5.3334 -5.8441 5.6379 -1.2256
Sample no. 28:
Input data: 3.0600 4.4000
Target: 0.5700 0.5500 0.8500 0.9700 0.2600 3.0400
Output: 0.6096 0.5672 0.8841 1.0117 0.2550 3.1044
err(%): -6.9501 -3.1271 -4.0095 -4.2981 1.9353 -2.1184
Sample no. 29:
Input data: 2.6700 2.9900
Target: 0.7200 0.6100 1.0100 1.3100 0.2400 2.5800
Output: 0.6788 0.5979 0.9799 1.2613 0.2512 2.5587
err(%): 5.7245 1.9792 2.9824 3.7185 -4.6520 0.8264
Sample no. 30:
Input data: 3.2700 4.4000
Target: 0.6400 0.5800 0.9200 1.0000 0.2600 3.3100
Output: 0.6833 0.6003 0.9519 1.0513 0.2557 3.3515
err(%): -6.7702 -3.4935 -3.4669 -5.1260 1.6611 -1.2523
Sample no. 31:
Input data: 3.0700 2.7900
Target: 1.0500 0.7300 1.2800 1.5700 0.2500 3.0500
Output: 0.9791 0.7062 1.2407 1.5004 0.2501 3.0182
err(%): 6.7523 3.2617 3.0708 4.4344 -0.0517 1.0423
Sample no. 32:
Input data: 2.8600 3.8000
Target: 0.6100 0.5400 0.8800 1.0200 0.2500 2.8200
Output: 0.5730 0.5647 0.8782 1.0623 0.2591 2.8494
err(%): 6.0684 -4.5800 0.1995 -4.1432 -3.6388 -1.0436
Sample no. 33:
Input data: 3.0600 3.8000
Target: 0.6600 0.5700 0.9600 1.0600 0.2500 3.0700
Output: 0.6486 0.5964 0.9520 1.1080 0.2581 3.0900
err(%): 1.7255 -4.6245 0.8315 -4.5275 -3.2566 -0.6512
Sample no. 34:
Input data: 3.2800 3.3900
Target: 0.8200 0.6600 1.1400 1.2900 0.2600 3.3400
Output: 0.8403 0.6661 1.1298 1.2807 0.2549 3.3447
err(%): -2.4698 -0.9228 0.8947 0.7242 1.9683 -0.1401
Sample no. 35:
Input data: 2.8600 4.2000
Target: 0.5100 0.5500 0.8100 0.9700 0.2600 2.8400
Output: 0.5486 0.5451 0.8348 0.9965 0.2574 2.8674
err(%): -7.5597 0.8977 -3.0625 -2.7327 1.0010 -0.9653
Sample no. 36:
Input data: 2.6500 3.6000
Target: 0.5100 0.5300 0.8500 1.0400 0.2500 2.6000
Output: 0.5245 0.5436 0.8312 1.0558 0.2599 2.5906
err(%): -2.8335 -2.5583 2.2114 -1.5238 -3.9534 0.3600

Appendix c The training results of the test sequential rig

The training data sets of the test sequential rig and the outputs of the trained neural network with the output errors are listed as follows. (The outputs of the rig and the trained neural network are plotted in Figure 5.30 to Figure 5.33.)

Sample no. 1
Input data: 0.0625
Target: 3.7703 0.0001 3.5367 0.0001
Output: 3.7914 -0.0070 3.5250 0.0138
error: 0.0211 -0.0071 -0.0117 0.0137
Sample no. 2
Input data: 0.1250
Target: 3.7690 0.0001 3.5333 0.0001
Output: 3.7847 -0.0050 3.5343 0.0109
error: 0.0157 -0.0051 0.0010 0.0108
Sample no. 3
Input data: 0.2500
Target: 3.7690 0.0001 3.5350 0.0001
Output: 3.7705 0.0043 3.5417 -0.0007
error: 0.0015 0.0042 0.0067 -0.0008
Sample no. 4
Input data: 0.3750
Target: 3.7690 0.0001 3.5367 0.0001
Output: 3.7550 0.0138 3.5440 -0.0125
error: -0.0140 0.0137 0.0073 -0.0126
Sample no. 5
Input data: 0.5000
Target: 3.7720 0.0001 3.5383 0.0001
Output: 3.7387 0.0159 3.5453 -0.0158
error: -0.0333 0.0158 0.0070 -0.0159
Sample no. 6
Input data: 0.6250
Target: 3.7657 0.0001 3.5317 0.0001
Output: 3.7250 0.0062 3.5412 -0.0037
error: -0.0407 0.0061 0.0095 -0.0038
Sample no. 7
Input data: 0.7500
Target: 3.7707 0.0001 3.5333 0.0001
Output: 3.7602 -0.0198 3.5417 0.0205
error: -0.0105 -0.0199 0.0084 0.0204
Sample no. 8
Input data: 0.8125
Target: 3.7673 0.0001 3.5350 0.0001
Output: 3.9434 0.0095 3.4709 -0.0078
error: 0.1761 0.0094 -0.0641 -0.0079
Sample no. 9
Input data: 0.8438
Target: 3.7393 0.0001 2.7910 0.0001
Output: 3.5432 -0.0269 2.8624 -0.0086
error: -0.1961 -0.0270 0.0714 -0.0087
Sample no. 10
Input data: 0.8594
Target: 2.0447 0.0001 2.5483 0.0001
Output: 2.1378 0.0048 2.5086 -0.0041
error: 0.0931 0.0047 -0.0397 -0.0042
Sample no. 11
Input data: 0.8750
Target: 0.8861 0.2409 2.3060 0.0001
Output: 0.9104 0.2828 2.3008 0.0294
error: 0.0243 0.0419 -0.0052 0.0293
Sample no. 12
Input data: 0.8906
Target: 0.6045 0.4167 2.0683 0.0001
Output: 0.5924 0.3612 2.0992 0.0040
error: -0.0121 -0.0555 0.0309 0.0039
Sample no. 13
Input data: 0.9375
Target: 0.5882 0.3353 1.8663 0.0001
Output: 0.4989 0.3261 1.8340 -0.0357
error: -0.0893 -0.0092 -0.0323 -0.0358
Sample no. 14
Input data: 1.0000
Target: 0.5118 0.3060 1.7620 0.0001
Output: 0.4837 0.2884 1.7527 -0.0132
error: -0.0281 -0.0176 -0.0093 -0.0133
Sample no. 15
Input data: 1.0630
Target: 0.4662 0.2686 1.7280 0.0001
Output: 0.4727 0.2825 1.7239 0.0031
error: 0.0065 0.0139 -0.0041 0.0030
Sample no. 16
Input data: 1.1250
Target: 0.4564 0.2653 1.7070 0.0001
Output: 0.4656 0.2948 1.7170 0.0072
error: 0.0092 0.0295 0.0100 0.0071
Sample no. 17
Input data: 1.1880
Target: 0.4515 0.2604 1.7007 0.0001
Output: 0.4622 0.3243 1.7301 0.0040
error: 0.0107 0.0639 0.0294 0.0039
Sample no. 18
Input data: 1.2500
Target: 0.4482 0.2604 1.6973 0.0001
Output: 0.4610 0.3610 1.7655 0.0082
error: 0.0128 0.1006 0.0682 0.0081
Sample no. 19
Input data: 1.3130
Target: 0.2482 0.6137 1.9910 0.0017
Output: 0.4578 0.3900 1.8285 0.0481
error: 0.2096 -0.2237 -0.1625 0.0464
Sample no. 20
Input data: 1.3750
Target: 0.5308 0.4024 2.0170 0.1611
Output: 0.4462 0.4244 1.9572 0.1563
error: -0.0846 0.0220 -0.0598 -0.0048
Sample no. 21
Input data: 1.4380
Target: 0.5568 0.3563 1.8127 0.4280
Output: 0.4241 0.4004 2.1114 0.3618
error: -0.1327 0.0441 0.2987 -0.0662
Sample no. 22
Input data: 1.5000
Target: 0.3848 0.3001 2.3677 0.6429
Output: 0.3984 0.3525 2.2193 0.6152
error: 0.0136 0.0524 -0.1484 -0.0277
Sample no. 23
Input data: 1.5630
Target: 0.3571 0.3783 2.3940 0.8056
Output: 0.3775 0.3135 2.3569 0.8642
error: 0.0204 -0.0648 -0.0371 0.0586
Sample no. 24
Input data: 1.7500
Target: 0.3767 0.3295 2.4137 1.3854
Output: 0.3855 0.3149 2.4427 1.3989
error: 0.0088 -0.0146 0.0290 0.0135
Sample no. 25
Input data: 1.8130
Target: 0.4580 0.3371 2.4087 1.5640
Output: 0.4063 0.3268 2.4179 1.5711
error: -0.0517 -0.0103 0.0092 0.0071
Sample no. 26
Input data: 1.8750
Target: 0.4483 0.3044 2.3823 1.7674
Output: 0.4321 0.3382 2.3956 1.7498
error: -0.0162 0.0338 0.0133 -0.0176
Sample no. 27
Input data: 1.9380
Target: 0.4092 0.3181 2.4000 1.9630
Output: 0.4558 0.3451 2.3785 1.9438
error: 0.0466 0.0270 -0.0215 -0.0192
Sample no. 28
Input data: 2.0000
Target: 0.4255 0.3874 2.3877 2.1454
Output: 0.4665 0.3541 2.3782 2.1408
error: 0.0410 -0.0333 -0.0095 -0.0046
Sample no. 29
Input data: 2.0630

Target: 0.4596 0.3629 2.3860 2.3307
Output: 0.4575 0.3686 2.3864 2.3440
error: -0.0021 0.0057 0.0004 0.0133
Sample no. 30
Input data: 2.1250
Target: 0.4434 0.3832 2.3923 2.5390
Output: 0.4249 0.3766 2.3928 2.5490
error: -0.0185 -0.0066 0.0005 0.0100
Sample no. 31
Input data: 2.3130
Target: 0.4418 0.3385 2.3953 3.1234
Output: 0.3576 0.3485 2.4856 3.1459
error: -0.0842 0.0100 0.0903 0.0225
Sample no. 32
Input data: 2.3280
Target: 0.4222 0.3337 2.6397 3.2277
Output: 0.4859 0.3421 2.4844 3.1818
error: 0.0637 0.0084 -0.1553 -0.0459
Sample no. 33
Input data: 2.3440
Target: 0.6290 0.3955 2.3860 3.2180
Output: 0.7422 0.3739 2.4838 3.2296
error: 0.1132 -0.0216 0.0978 0.0116
Sample no. 34
Input data: 2.3590
Target: 1.2701 0.4752 2.3970 3.2717
Output: 1.1162 0.4631 2.4024 3.2765
error: -0.1539 -0.0121 0.0054 0.0048
Sample no. 35
Input data: 2.3750
Target: 1.6463 0.6592 2.4020 3.3124
Output: 1.6996 0.6857 2.3450 3.3169
error: 0.0533 0.0265 -0.0570 0.0045
Sample no. 36
Input data: 2.3910
Target: 2.3167 1.0093 2.4167 3.3497
Output: 2.3492 1.0394 2.4142 3.3436
error: 0.0325 0.0301 -0.0025 -0.0061
Sample no. 37
Input data: 2.4060
Target: 3.0300 1.5167 2.4883 3.3774
Output: 2.9506 1.4740 2.5466 3.3797
error: -0.0794 -0.0427 0.0583 0.0023
Sample no. 38
Input data: 2.4380
Target: 3.7867 2.5054 3.0677 3.4227
Output: 3.8526 2.5295 3.0310 3.4202
error: 0.0659 0.0241 -0.0367 -0.0025
Sample no. 39
Input data: 2.5000
Target: 4.6243 3.3960 3.4130 3.5044
Output: 4.5632 3.3532 3.4274 3.5061
error: -0.0611 -0.0428 0.0144 0.0017
Sample no. 40
Input data: 2.5630
Target: 4.6060 3.3470 3.4033 3.5777
Output: 4.6561 3.3749 3.3814 3.5843
error: 0.0501 0.0279 -0.0219 0.0066
Sample no. 41
Input data: 2.6880
Target: 4.6343 3.3310 3.4070 3.7240
Output: 4.6092 3.3393 3.4326 3.7131
error: -0.0251 0.0083 0.0256 -0.0109
Sample no. 42
Input data: 2.9380
Target: 4.6347 3.3567 3.3970 3.9990
Output: 4.6510 3.3465 3.3821 4.0084
error: 0.0163 -0.0102 -0.0149 0.0094
Sample no. 43
Input data: 3.3130
Target: 4.6230 3.3650 3.3990 4.3980
Output: 4.6064 3.3689 3.3877 4.4005
error: -0.0166 0.0039 -0.0113 0.0025
Sample no. 44
Input data: 3.3750
Target: 4.5257 3.3640 3.4013 4.4567
Output: 4.5482 3.3658 3.3961 4.4628
error: 0.0225 0.0018 -0.0052 0.0061
Sample no. 45
Input data: 3.4380
Target: 4.4830 3.3624 3.3913 4.5167
Output: 4.4706 3.3639 3.4042 4.5196
error: -0.0124 0.0015 0.0129 0.0029
Sample no. 46
Input data: 3.6250
Target: 4.1627 3.3737 3.4130 4.6810
Output: 4.1712 3.3663 3.4187 4.6731
error: 0.0085 -0.0074 0.0057 -0.0079
Sample no. 47
Input data: 3.7500
Target: 3.9560 3.3757 3.4080 4.7934
Output: 3.9466 3.3763 3.4144 4.7804
error: -0.0094 0.0006 0.0064 -0.0130
Sample no. 48
Input data: 4.1250
Target: 3.3277 3.3790 3.4127 5.1060
Output: 3.3313 3.3895 3.4125 5.1132
error: 0.0036 0.0105 -0.0002 0.0072
Sample no. 49
Input data: 4.5000
Target: 2.7690 3.3904 3.4193 5.4200
Output: 2.7669 3.3727 3.3996 5.4155
error: -0.0021 -0.0177 -0.0197 -0.0045
Sample no. 50
Input data: 4.8750
Target: 2.0983 3.3724 3.4097 5.7360
Output: 2.1161 3.3895 3.4409 5.7400
error: 0.0178 0.0171 0.0312 0.0040
Sample no. 51
Input data: 5.2500
Target: 1.4833 3.3614 3.4013 6.0777
Output: 1.4033 3.2902 3.1917 6.0735
error: -0.0800 -0.0712 -0.2096 -0.0042
Sample no. 52
Input data: 5.3750
Target: 1.2264 3.2922 2.8321 6.1716
Output: 1.3481 3.3441 3.0579 6.1513
error: 0.1217 0.0519 0.2258 -0.0203
Sample no. 53
Input data: 5.5000
Target: 0.9495 3.0014 2.0700 6.2550
Output: 0.8855 3.0096 1.9697 6.2899
error: -0.0640 0.0082 -0.1003 0.0349
Sample no. 54
Input data: 5.5160
Target: 0.6615 2.6950 1.4417 6.2550
Output: 0.6295 2.7598 1.5801 6.2641
error: -0.0320 0.0648 0.1384 0.0091
Sample no. 55
Input data: 5.5310
Target: 0.2448 2.5407 1.2743 6.2457
Output: 0.3259 2.4915 1.1958 6.2377
error: 0.0811 -0.0492 -0.0785 -0.0080
Sample no. 56
Input data: 5.7500
Target: 0.2644 2.3324 1.0917 6.1810
Output: 0.3029 2.3552 1.0852 6.1758
error: 0.0385 0.0228 -0.0065 -0.0052
Sample no. 57
Input data: 5.8750
Target: 0.4303 2.3277 1.0900 6.1060
Output: 0.3682 2.2961 1.1005 6.1028
error: -0.0621 -0.0316 0.0105 -0.0032
Sample no. 58
Input data: 5.9380
Target: 0.4255 2.2574 1.0850 6.0784
Output: 0.3984 2.2746 1.1000 6.0829
error: -0.0271 0.0172 0.0150 0.0045
Sample no. 59
Input data: 6.1250
Target: 0.4320 2.1877 1.0903 6.0040
Output: 0.4393 2.1828 1.0780 6.0143

error: 0.0073 -0.0049 -0.0123 0.0103
Sample no. 60
Input data: 6.2500
Target: 0.4352 2.1420 1.0853 5.9437
Output: 0.4397 2.1179 1.0770 5.9409
error: 0.0045 -0.0241 -0.0083 -0.0028
Sample no. 61
Input data: 6.3130
Target: 0.4352 2.1047 1.0933 5.9160
Output: 0.4390 2.0920 1.0811 5.9086
error: 0.0038 -0.0127 -0.0122 -0.0074
Sample no. 62
Input data: 6.3750
Target: 0.4369 2.0770 1.0837 5.8884
Output: 0.4390 2.0684 1.0871 5.8811
error: 0.0021 -0.0086 0.0034 -0.0073
Sample no. 63
Input data: 6.4380
Target: 0.4385 2.0474 1.0900 5.8577
Output: 0.4400 2.0456 1.0923 5.8558
error: 0.0015 -0.0018 0.0023 -0.0019
Sample no. 64
Input data: 6.5000
Target: 0.4352 2.0180 1.0883 5.8297
Output: 0.4417 2.0225 1.0974 5.8312
error: 0.0065 0.0045 0.0091 0.0015
Sample no. 65
Input data: 6.5630
Target: 0.4385 1.9840 1.1013 5.8037
Output: 0.4439 1.9970 1.1016 5.8053
error: 0.0054 0.0130 0.0003 0.0016
Sample no. 66
Input data: 6.6250
Target: 0.4336 1.9564 1.0997 5.7794
Output: 0.4464 1.9687 1.1057 5.7785
error: 0.0128 0.0123 0.0060 -0.0009
Sample no. 67
Input data: 6.8750
Target: 0.4401 1.8310 1.0963 5.6734
Output: 0.4520 1.8339 1.1063 5.6681
error: 0.0119 0.0029 0.0100 -0.0053
Sample no. 68
Input data: 7.1250
Target: 0.4450 1.6977 1.1047 5.5837
Output: 0.4351 1.6911 1.0826 5.5855
error: -0.0099 -0.0066 -0.0221 0.0018
Sample no. 69
Input data: 7.3750
Target: 0.4483 1.5904 1.1063 5.4964
Output: 0.4306 1.5950 1.1148 5.4999
error: -0.0177 0.0046 0.0085 0.0035
Sample no. 70
Input data: 7.6250
Target: 0.4483 1.4617 1.1113 5.4164
Output: 0.4665 1.4666 1.1162 5.4156
error: 0.0182 0.0049 0.0049 -0.0008
Sample no. 71
Input data: 7.9380
Target: 0.4564 1.3427 1.1097 5.3187
Output: 0.4573 1.3262 1.1021 5.3235
error: 0.0009 -0.0165 -0.0076 0.0048
Sample no. 72
Input data: 8.1880
Target: 0.4613 1.2777 1.1047 5.2557
Output: 0.4630 1.3004 1.1136 5.2462
error: 0.0017 0.0227 0.0089 -0.0095
Sample no. 73
Input data: 8.4380
Target: 0.4678 1.2027 1.1147 5.1754
Output: 0.4632 1.1876 1.1097 5.1797
error: -0.0046 -0.0151 -0.0050 0.0043
Sample no. 74
Input data: 8.6880
Target: 0.4662 1.1099 1.1097 5.1104
Output: 0.4571 1.1201 1.0997 5.1125
error: -0.0091 0.0102 -0.0100 0.0021
Sample no. 75
Input data: 8.9380
Target: 0.4694 1.0271 1.1193 5.0470
Output: 0.4804 1.0111 1.1412 5.0449
error: 0.0110 -0.0160 0.0219 -0.0021
Sample no. 76
Input data: 9.1880
Target: 0.4678 0.9505 1.1210 4.9934
Output: 0.4672 0.9566 1.1194 4.9973
error: -0.0006 0.0061 -0.0016 0.0039
Sample no. 77
Input data: 9.4380
Target: 0.4727 0.8691 1.1273 4.9394
Output: 0.4957 0.8888 1.1212 4.9448
error: 0.0230 0.0197 -0.0061 0.0054
Sample no. 78
Input data: 9.6880
Target: 0.4727 0.8024 1.1227 4.8794
Output: 0.4571 0.7733 1.1085 4.8589
error: -0.0156 -0.0291 -0.0142 -0.0205
Sample no. 79
Input data: 9.9380
Target: 0.3945 0.6202 1.0867 4.7300
Output: 0.3695 0.5922 1.0726 4.7458
error: -0.0250 -0.0280 -0.0141 0.0158
Sample no. 80
Input data: 10.0000
Target: 0.3669 0.4883 1.0673 4.6860
Output: 0.3662 0.5485 1.0849 4.6725
error: -0.0007 0.0602 0.0176 -0.0135
Sample no. 81
Input data: 10.0600
Target: 0.3311 0.4720 1.0397 4.6257
Output: 0.3622 0.4845 1.0651 4.6396
error: 0.0311 0.0125 0.0254 0.0139
Sample no. 82
Input data: 10.1300
Target: 0.3441 0.4541 1.0316 4.5640
Output: 0.3573 0.4293 1.0432 4.5717
error: 0.0132 -0.0248 0.0116 0.0077
Sample no. 83
Input data: 10.1900
Target: 0.3701 0.4102 1.0477 4.5054
Output: 0.3639 0.3923 1.0344 4.5105
error: -0.0062 -0.0179 -0.0133 0.0051
Sample no. 84
Input data: 10.2500
Target: 0.3913 0.3678 1.0657 4.4517
Output: 0.3864 0.3618 1.0487 4.4487
error: -0.0049 -0.0060 -0.0170 -0.0030
Sample no. 85
Input data: 10.3100
Target: 0.4238 0.3320 1.0967 4.4014
Output: 0.4135 0.3364 1.0741 4.3942
error: -0.0103 0.0044 -0.0226 -0.0072
Sample no. 86
Input data: 10.3800
Target: 0.4531 0.3092 1.1193 4.3607
Output: 0.4518 0.3170 1.1187 4.3398
error: -0.0013 0.0078 -0.0006 -0.0209
Sample no. 87
Input data: 10.4400
Target: 0.4825 0.3027 1.1487 4.3230
Output: 0.4811 0.3067 1.1548 4.3109
error: -0.0014 0.0040 0.0061 -0.0121
Sample no. 88
Input data: 10.5000
Target: 0.5199 0.2913 1.1877 4.2807
Output: 0.5154 0.3062 1.1936 4.2897
error: -0.0045 0.0149 0.0059 0.0090
Sample no. 89
Input data: 10.5600
Target: 0.5622 0.3174 1.2170 4.2417
Output: 0.5566 0.3137 1.2339 4.2615
error: -0.0056 -0.0037 0.0169 0.0198
Sample no. 90

Input data: 10.6300
Target: 0.6094 0.3467 1.2723 4.1994
Output: 0.6127 0.3380 1.2808 4.2084
error: 0.0033 -0.0087 0.0085 0.0090
Sample no. 91
Input data: 10.6900
Target: 0.6534 0.3662 1.3127 4.1537
Output: 0.6563 0.3647 1.3196 4.1503
error: 0.0029 -0.0015 0.0069 -0.0034
Sample no. 92
Input data: 10.7500
Target: 0.6924 0.3988 1.3637 4.1000
Output: 0.6894 0.3912 1.3504 4.0947
error: -0.0030 -0.0076 -0.0133 -0.0053
Sample no. 93
Input data: 10.8100
Target: 0.7168 0.4265 1.3827 4.0447
Output: 0.7147 0.4195 1.3763 4.0395
error: -0.0021 -0.0070 -0.0064 -0.0052
Sample no. 94
Input data: 10.8800
Target: 0.7120 0.4362 1.3847 3.9860
Output: 0.7313 0.4475 1.3891 3.9839
error: 0.0193 0.0113 0.0044 -0.0021
Sample no. 95
Input data: 11.2500
Target: 0.7136 0.4378 1.3813 3.6017
Output: 0.6997 0.4436 1.3649 3.6053
error: -0.0139 0.0058 -0.0164 0.0036
Sample no. 96
Input data: 11.5600
Target: 0.7119 0.4297 1.3797 3.2797
Output: 0.7086 0.4150 1.3985 3.2677
error: -0.0033 -0.0147 0.0188 -0.0120
Sample no. 97
Input data: 11.8800
Target: 0.7315 0.4476 1.3797 2.9560
Output: 0.7266 0.4552 1.3680 2.9693
error: -0.0049 0.0076 -0.0117 0.0133
Sample no. 98
Input data: 12.1300
Target: 0.7266 0.4427 1.3880 2.6937
Output: 0.7265 0.4185 1.3897 2.6833
error: -0.0001 -0.0242 0.0017 -0.0104
Sample no. 99
Input data: 12.4400
Target: 0.7298 0.4459 1.3797 2.3647
Output: 0.7606 0.4820 1.3853 2.3839
error: 0.0308 0.0361 0.0056 0.0192
Sample no. 100
Input data: 12.7500
Target: 0.7331 0.4476 1.3783 2.0477
Output: 0.7301 0.4573 1.3789 2.0207
error: -0.0030 0.0097 0.0006 -0.0270
Sample no. 101
Input data: 13.0600
Target: 0.7282 0.4443 1.3813 1.7287
Output: 0.7043 0.4282 1.3714 1.7361
error: -0.0239 -0.0161 -0.0099 0.0074
Sample no. 102
Input data: 13.3800
Target: 0.7299 0.4411 1.3863 1.4064
Output: 0.7141 0.4339 1.3851 1.4307
error: -0.0158 -0.0072 -0.0012 0.0243
Sample no. 103
Input data: 13.6900
Target: 0.7380 0.4395 1.3817 1.0854
Output: 0.7600 0.4399 1.3983 1.0699
error: 0.0220 0.0004 0.0166 -0.0155
Sample no. 104
Input data: 14.0000
Target: 0.7331 0.4362 1.3960 0.7633
Output: 0.7336 0.4447 1.3910 0.7609
error: 0.0005 0.0085 -0.0050 -0.0024
Sample no. 105
Input data: 14.5600
Target: 0.7380 0.4525 1.3943 0.1839
Output: 0.7218 0.4718 1.3819 0.1971
error: -0.0162 0.0193 -0.0124 0.0132
Sample no. 106
Input data: 14.6300
Target: 0.7640 0.5306 1.3927 0.1188
Output: 0.8375 0.4771 1.3914 0.1111
error: 0.0735 -0.0535 -0.0013 -0.0077
Sample no. 107
Input data: 14.6900
Target: 1.0065 0.6122 1.5897 0.0667
Output: 1.0607 0.6199 1.6430 0.0537
error: 0.0542 0.0077 0.0533 -0.0130
Sample no. 108
Input data: 14.7000
Target: 1.1464 0.6359 1.7053 0.0618
Output: 1.1120 0.6518 1.6757 0.0670
error: -0.0344 0.0159 -0.0296 0.0052
Sample no. 109
Input data: 14.7200
Target: 1.3191 0.7402 1.8633 0.0472
Output: 1.2981 0.7672 1.8916 0.0521
error: -0.0210 0.0270 0.0283 0.0049
Sample no. 110
Input data: 14.7300
Target: 1.5357 0.8052 2.0407 0.0423
Output: 1.4118 0.8150 1.9680 0.0435
error: -0.1239 0.0098 -0.0727 0.0012
Sample no. 111
Input data: 14.7500
Target: 1.7877 0.8510 2.2670 0.0293
Output: 1.8587 0.8997 2.3377 0.0338
error: 0.0710 0.0487 0.0707 0.0045
Sample no. 112
Input data: 14.7700
Target: 2.0437 0.9447 2.5257 0.0244
Output: 2.1510 0.8335 2.5329 0.0139
error: 0.1073 -0.1112 0.0072 -0.0105
Sample no. 113
Input data: 14.8100
Target: 3.0297 0.4680 3.1797 0.0033
Output: 2.8648 0.4968 3.0994 -0.0024
error: -0.1649 0.0288 -0.0803 -0.0057
Sample no. 114
Input data: 14.8800
Target: 3.4187 0.0001 3.3880 0.0001
Output: 3.3766 -0.0005 3.4156 -0.0178
error: -0.0421 -0.0006 0.0276 -0.0179
Sample no. 115
Input data: 14.9400
Target: 3.3713 0.0001 3.3723 0.0001
Output: 3.4144 -0.0244 3.3868 0.0229
error: 0.0431 -0.0245 0.0145 0.0228
Sample no. 116
Input data: 15.2500
Target: 3.3797 0.0001 3.3817 0.0001
Output: 3.5366 0.0031 3.3930 0.0153
error: 0.1569 0.0030 0.0113 0.0152
Sample no. 117
Input data: 15.5600
Target: 3.7653 0.0001 3.5330 0.0001
Output: 3.6570 0.0357 3.4864 -0.0274
error: -0.1083 0.0356 -0.0466 -0.0275
Sample no. 118
Input data: 15.8800
Target: 3.7703 0.0001 3.5367 0.0001
Output: 3.7678 -0.0101 3.5631 0.0162
error: -0.0025 -0.0102 0.0264 0.0161

Appendix D

Fuzzy diagnostic outputs by fuzzy intersection operation

***** data no:1 *****
Symptoms: 2.000 0.000 3.150 1.622 1.947 1.594 0.202 2.115
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.295 0.000 0.000 0.531
0.000 0.000 0.000 0.000 0.000 0.000 0.021 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.295
***** data no:2 *****
Symptoms: 2.000 0.000 3.060 1.469 1.738 1.466 0.273 2.069
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.310 0.000 0.000 0.489
0.000 0.000 0.000 0.000 0.000 0.000 0.009 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.310
***** data no:3 *****
Symptoms: 2.000 0.000 3.102 1.565 2.022 1.652 0.088 2.095
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.302 0.000 0.000 0.522
0.000 0.000 0.000 0.000 0.000 0.000 0.015 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.302
***** data no:4 *****
Symptoms: 2.000 0.000 3.125 1.495 1.872 1.609 0.129 2.051
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.316 0.000 0.000 0.498
0.000 0.000 0.000 0.000 0.000 0.000 0.018 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.316
***** data no:5 *****
Symptoms: 2.000 0.000 2.994 1.635 2.274 1.838 -0.034 2.035
Faulty group=5 Most suitable rule no=9
0.002 0.000 0.000 0.000 0.000 0.000 0.242 0.000 0.000 0.545
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.011 0.322
***** data no:6 *****
Symptoms: 2.000 0.000 3.041 1.679 2.173 1.691 0.216 2.062
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.276 0.000 0.000 0.560
0.000 0.000 0.000 0.000 0.000 0.000 0.006 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.313
***** data no:7 *****
Symptoms: 2.000 0.000 3.032 1.615 2.010 1.400 0.148 2.033
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.322 0.000 0.000 0.467
0.000 0.000 0.000 0.000 0.000 0.000 0.005 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.322
***** data no:8 *****
Symptoms: 2.000 0.000 3.256 1.467 1.921 1.506 0.097 2.127
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.291 0.000 0.000 0.489
0.000 0.000 0.000 0.000 0.000 0.000 0.037 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.291
***** data no:9 *****
Symptoms: 2.000 0.000 3.335 1.424 1.892 1.499 -0.136 2.093
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.302 0.000 0.000 0.475
0.000 0.000 0.000 0.000 0.000 0.000 0.048 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.045 0.302
***** data no:10 *****
Symptoms: 2.000 0.000 2.728 1.346 2.003 1.730 0.030 1.926
Faulty group=5 Most suitable rule no=9
0.091 0.000 0.000 0.000 0.000 0.000 0.332 0.000 0.000 0.449
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:11 *****
Symptoms: -2.000 0.000 -3.054 -1.403 -1.742 -1.415 -0.066 -2.029
Faulty group=5 Most suitable rule no=8
0.000 0.333 0.000 0.000 0.324 0.000 0.000 0.000 0.468 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.008 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:12 *****
Symptoms: -2.000 0.000 -3.124 -1.450 -1.813 -1.461 -0.075 -2.081
Faulty group=5 Most suitable rule no=8
0.000 0.333 0.000 0.000 0.306 0.000 0.000 0.000 0.483 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.018 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:13 *****
Symptoms: -2.000 0.000 -3.147 -1.545 -1.922 -1.672 0.193 -2.027
Faulty group=5 Most suitable rule no=8
0.000 0.333 0.000 0.000 0.324 0.000 0.000 0.000 0.515 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.021 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.064 0.000 0.000
***** data no:14 *****
Symptoms: -2.000 0.000 -3.025 -1.492 -1.818 -1.610 0.208 -2.047
Faulty group=5 Most suitable rule no=8
0.000 0.333 0.000 0.000 0.318 0.000 0.000 0.000 0.497 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.004 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.069 0.000 0.000
***** data no:15 *****
Symptoms: -2.000 0.000 -2.909 -1.485 -1.863 -1.562 -0.083 -2.005
Faulty group=5 Most suitable rule no=8
0.030 0.333 0.030 0.030 0.332 0.030 0.000 0.000 0.495 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:16 *****
Symptoms: -2.000 0.000 -2.703 -1.312 -1.854 -1.503 0.164 -1.912
Faulty group=5 Most suitable rule no=8
0.099 0.333 0.099 0.099 0.333 0.099 0.000 0.000 0.437 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.055 0.000 0.000
***** data no:17 *****
Symptoms: -2.000 0.000 -2.559 -1.210 -1.949 -1.677 0.191 -1.932
Faulty group=5 Most suitable rule no=8
0.147 0.333 0.147 0.147 0.333 0.147 0.000 0.000 0.403 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.064 0.000 0.000
***** data no:18 *****
Symptoms: -2.000 0.000 -2.558 -1.340 -2.004 -1.642 0.073 -1.915
Faulty group=5 Most suitable rule no=8
0.147 0.333 0.147 0.147 0.333 0.147 0.000 0.000 0.447 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.024 0.000 0.000
***** data no:19 *****
Symptoms: -2.000 0.000 -2.567 -1.578 -2.149 -1.661 -0.033 -1.908
Faulty group=5 Most suitable rule no=8
0.144 0.333 0.144 0.144 0.333 0.144 0.000 0.000 0.526 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:20 *****
Symptoms: -2.000 0.000 -2.726 -1.572 -2.066 -1.766 -0.113 -1.904
Faulty group=5 Most suitable rule no=8
0.091 0.333 0.091 0.091 0.333 0.091 0.000 0.000 0.524 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:21 *****
Symptoms: 0.000 2.000 -1.977 -1.585 -2.263 -2.761 0.335 0.081
Faulty group=6 Most suitable rule no=11
0.080 0.000 0.000 0.000 0.333 0.080 0.000 0.000 0.000 0.000
0.000 0.528 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.080 0.000 0.000
***** data no:22 *****
Symptoms: 0.000 2.000 -2.082 -1.769 -2.300 -2.835 0.259 0.023
Faulty group=6 Most suitable rule no=11
0.055 0.000 0.000 0.000 0.333 0.055 0.000 0.000 0.000 0.000
0.000 0.590 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.055 0.000 0.000
***** data no:23 *****
Symptoms: 0.000 2.000 -2.003 -1.642 -2.157 -2.700 0.177 0.049
Faulty group=6 Most suitable rule no=11
0.100 0.000 0.000 0.000 0.333 0.100 0.000 0.000 0.000 0.000
0.000 0.547 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.059 0.000 0.000
***** data no:24 *****
Symptoms: 0.000 2.000 -2.033 -1.719 -2.161 -2.720 0.138 -0.005
Faulty group=6 Most suitable rule no=11
0.093 0.002 0.002 0.002 0.333 0.093 0.000 0.000 0.000 0.000
0.000 0.573 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.046 0.000 0.000
***** data no:25 *****
Symptoms: 0.000 2.000 -1.963 -1.507 -1.884 -2.470 0.006 0.027
Faulty group=6 Most suitable rule no=11

0.177 0.000 0.000 0.000 0.333 0.177 0.000 0.000 0.000 0.000
0.000 0.502 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.002 0.000 0.000
***** data no:26 *****
Symptoms: 0.000 2.000 -1.714 -1.300 -1.770 -2.271 0.158 0.015
Faulty group=6 Most suitable rule no=11
0.243 0.000 0.000 0.000 0.333 0.243 0.000 0.000 0.000 0.000
0.000 0.433 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.053 0.000 0.000
***** data no:27 *****
Symptoms: 0.000 2.000 -1.531 -1.205 -1.966 -2.364 0.221 -0.039
Faulty group=6 Most suitable rule no=11
0.212 0.013 0.013 0.013 0.333 0.212 0.000 0.000 0.000 0.000
0.000 0.402 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.074 0.000 0.000
***** data no:28 *****
Symptoms: 0.000 2.000 -1.453 -1.500 -2.075 -2.595 0.064 0.105
Faulty group=6 Most suitable rule no=11
0.135 0.000 0.000 0.000 0.333 0.135 0.000 0.000 0.000 0.000
0.000 0.484 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.021 0.000 0.000
***** data no:29 *****
Symptoms: 0.000 2.000 -1.723 -1.577 -2.030 -2.764 -0.107 0.053
Faulty group=6 Most suitable rule no=11
0.079 0.000 0.000 0.000 0.333 0.079 0.000 0.000 0.000 0.000
0.000 0.526 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:30 *****
Symptoms: 0.000 2.000 -1.742 -1.554 -2.074 -2.432 -0.027 -0.072
Faulty group=6 Most suitable rule no=11
0.189 0.024 0.024 0.024 0.333 0.189 0.000 0.000 0.000 0.000
0.000 0.518 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:31 *****
Symptoms: 0.000 -2.000 1.855 1.585 1.842 2.410 0.252 0.079
Faulty group=6 Most suitable rule no=10
0.197 0.000 0.000 0.000 0.000 0.000 0.333 0.000 0.000 0.000
0.528 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:32 *****
Symptoms: 0.000 -2.000 1.766 1.410 1.620 2.274 0.337 0.040
Faulty group=6 Most suitable rule no=10
0.242 0.000 0.000 0.000 0.000 0.000 0.333 0.000 0.000 0.000
0.470 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:33 *****
Symptoms: 0.000 -2.000 1.840 1.504 1.906 2.473 0.140 0.051
Faulty group=6 Most suitable rule no=10
0.176 0.000 0.000 0.000 0.000 0.000 0.333 0.000 0.000 0.000
0.501 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:34 *****
Symptoms: 0.000 -2.000 1.862 1.414 1.738 2.446 0.177 -0.004
Faulty group=6 Most suitable rule no=10
0.185 0.000 0.000 0.001 0.000 0.000 0.333 0.000 0.000 0.000
0.471 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:35 *****
Symptoms: 0.000 -2.000 1.755 1.566 2.187 2.668 -0.015 0.014
Faulty group=6 Most suitable rule no=10
0.111 0.000 0.000 0.000 0.000 0.000 0.271 0.000 0.000 0.000
0.522 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.005 0.333
***** data no:36 *****
Symptoms: 0.000 -2.000 1.665 1.646 2.127 2.573 0.210 -0.023
Faulty group=6 Most suitable rule no=10
0.142 0.000 0.000 0.008 0.000 0.000 0.291 0.000 0.000 0.000
0.549 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:37 *****
Symptoms: 0.000 -2.000 0.057 0.190 -0.023 -0.055 0.200 -0.050
Faulty group=6 Most suitable rule no=0
0.333 0.000 0.008 0.017 0.000 0.008 0.000 0.008 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:38 *****

Symptoms: 0.000 -2.000 2.028 1.448 1.880 2.395 0.131 0.077
Faulty group=6 Most suitable rule no=10
0.202 0.000 0.000 0.000 0.000 0.000 0.333 0.000 0.000 0.000
0.483 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:39 *****
Symptoms: 0.000 -2.000 2.154 1.379 1.844 2.485 -0.114 0.025
Faulty group=6 Most suitable rule no=10
0.172 0.000 0.000 0.000 0.000 0.000 0.333 0.000 0.000 0.000
0.460 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.038 0.333
***** data no:40 *****
Symptoms: 0.000 -2.000 1.586 1.308 1.916 2.627 0.032 -0.106
Faulty group=6 Most suitable rule no=10
0.124 0.000 0.000 0.035 0.000 0.000 0.333 0.000 0.000 0.000
0.436 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.333
***** data no:41 *****
Symptoms: 0.000 0.000 -0.050 0.011 -0.219 -0.179 0.305 0.082
Faulty group=0 Most suitable rule no=0
0.898 0.000 0.000 0.000 0.000 0.073 0.000 0.004 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:42 *****
Symptoms: 0.000 0.000 -0.161 -0.181 -0.358 -0.296 0.311 0.032
Faulty group=0 Most suitable rule no=0
0.881 0.000 0.000 0.000 0.054 0.119 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.054 0.000 0.000
***** data no:43 *****
Symptoms: 0.000 0.000 -0.073 -0.062 -0.135 -0.119 0.164 0.051
Faulty group=0 Most suitable rule no=0
0.945 0.000 0.000 0.000 0.021 0.045 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.021 0.000 0.000
***** data no:44 *****
Symptoms: 0.000 0.000 -0.081 -0.154 -0.223 -0.143 0.163 -0.006
Faulty group=0 Most suitable rule no=0
0.926 0.002 0.002 0.002 0.027 0.074 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.027 0.000 0.000
***** data no:45 *****
Symptoms: 0.000 0.000 -0.088 0.039 0.153 0.101 -0.006 0.021
Faulty group=0 Most suitable rule no=0
0.949 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:46 *****
Symptoms: 0.000 0.000 -0.010 0.182 0.182 0.159 0.188 -0.003
Faulty group=0 Most suitable rule no=0
0.937 0.000 0.000 0.001 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:47 *****
Symptoms: 0.000 0.000 0.057 0.190 -0.023 -0.055 0.200 -0.050
Faulty group=0 Most suitable rule no=0
0.933 0.000 0.008 0.017 0.000 0.008 0.000 0.008 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:48 *****
Symptoms: 0.000 0.000 0.305 -0.019 -0.098 -0.094 0.102 0.092
Faulty group=0 Most suitable rule no=0
0.898 0.000 0.000 0.000 0.000 0.033 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:49 *****
Symptoms: 0.000 0.000 0.228 -0.095 -0.096 -0.142 -0.119 0.038
Faulty group=0 Most suitable rule no=0
0.924 0.000 0.000 0.000 0.000 0.032 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:50 *****
Symptoms: 0.000 0.000 -0.061 -0.116 -0.080 0.102 0.007 -0.090
Faulty group=0 Most suitable rule no=0
0.961 0.000 0.027 0.030 0.000 0.027 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.002 0.000 0.000
***** data no:51 *****
Symptoms: 0.000 0.000 -0.068 0.088 0.210 0.165 -0.043 -0.048
Faulty group=0 Most suitable rule no=0
0.930 0.000 0.000 0.016 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:52 *****
Symptoms: 0.000 0.000 -0.104 0.067 0.156 0.143 -0.088 -0.108
Faulty group=0 Most suitable rule no=0
0.948 0.000 0.000 0.036 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:53 *****
Symptoms: 0.000 0.000 -0.035 0.042 0.084 0.033 0.219 -0.090
Faulty group=0 Most suitable rule no=0
0.927 0.000 0.000 0.030 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:54 *****
Symptoms: 0.000 0.000 0.078 0.050 0.145 0.026 0.181 -0.066
Faulty group=0 Most suitable rule no=0
0.940 0.000 0.000 0.022 0.000 0.000 0.009 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.009
***** data no:55 *****
Symptoms: 0.000 0.000 0.102 -0.015 0.054 -0.006 -0.072 -0.008
Faulty group=0 Most suitable rule no=0
0.966 0.000 0.000 0.003 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:56 *****
Symptoms: 0.000 0.000 0.203 0.129 0.055 -0.008 0.184 0.072
Faulty group=0 Most suitable rule no=0
0.932 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:57 *****
Symptoms: 0.000 0.000 0.246 0.170 -0.006 -0.249 0.144 0.053
Faulty group=0 Most suitable rule no=0
0.917 0.000 0.000 0.000 0.000 0.002 0.000 0.002 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:58 *****
Symptoms: 0.000 0.000 0.084 0.050 -0.017 -0.093 0.045 0.025
Faulty group=0 Most suitable rule no=0
0.969 0.000 0.000 0.000 0.000 0.006 0.000 0.006 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:59 *****
Symptoms: 0.000 0.000 0.221 -0.133 -0.190 -0.077 -0.032 0.032
Faulty group=0 Most suitable rule no=0
0.926 0.000 0.000 0.000 0.000 0.063 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:60 *****
Symptoms: 0.000 0.000 0.211 -0.087 -0.145 -0.156 -0.158 0.056
Faulty group=0 Most suitable rule no=0
0.930 0.000 0.000 0.000 0.000 0.048 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:61 *****
Symptoms: 0.000 0.000 -0.982 -0.906 -1.412 -0.767 0.166 -1.354
Faulty group=1 Most suitable rule no=0
0.529 0.256 0.451 0.451 0.256 0.471 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.055 0.000 0.000
***** data no:62 *****
Symptoms: 0.000 0.000 -1.207 -1.381 -1.794 -1.448 0.039 -1.781
Faulty group=1 Most suitable rule no=2
0.402 0.402 0.517 0.402 0.402 0.406 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.013 0.000 0.000
***** data no:63 *****
Symptoms: 0.000 0.000 -1.202 -1.265 -1.548 -1.461 0.123 -3.036
Faulty group=1 Most suitable rule no=2

0.000 0.401 0.513 0.484 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:64 *****
Symptoms: 0.000 0.000 -0.442 -0.851 -1.300 -1.003 0.084 -1.209
Faulty group=1 Most suitable rule no=0
0.567 0.147 0.403 0.403 0.147 0.433 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.028 0.000 0.000
***** data no:65 *****
Symptoms: 0.000 0.000 -0.948 -1.098 -1.406 -1.129 0.082 -1.653
Faulty group=1 Most suitable rule no=3
0.449 0.316 0.469 0.531 0.316 0.449 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.027 0.000 0.000
***** data no:66 *****
Symptoms: 0.000 0.000 -2.217 -2.541 -3.181 -2.722 -0.147 -3.271
Faulty group=1 Most suitable rule no=1
0.000 0.739 0.093 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:67 *****
Symptoms: 0.000 0.000 -0.914 -1.053 -1.161 -0.565 -0.162 -1.485
Faulty group=1 Most suitable rule no=0
0.505 0.188 0.387 0.495 0.188 0.387 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:68 *****
Symptoms: 0.000 0.000 -1.096 -1.322 -1.817 -1.258 0.055 -1.932
Faulty group=1 Most suitable rule no=2
0.356 0.365 0.559 0.394 0.356 0.356 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.018 0.000 0.000
***** data no:69 *****
Symptoms: 0.000 0.000 -2.133 -2.515 -3.198 -2.594 0.231 -3.211
Faulty group=1 Most suitable rule no=1
0.000 0.711 0.135 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:70 *****
Symptoms: 0.000 0.000 -1.412 -2.020 -2.710 -2.117 -0.249 -2.549
Faulty group=1 Most suitable rule no=1
0.097 0.471 0.294 0.097 0.150 0.150 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:71 *****
Symptoms: 0.000 0.000 -0.802 -1.092 -1.314 -1.154 -0.319 -1.172
Faulty group=1 Most suitable rule no=0
0.562 0.267 0.391 0.391 0.267 0.438 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:72 *****
Symptoms: 0.000 0.000 -1.134 -1.597 -1.937 -1.610 0.083 -1.797
Faulty group=1 Most suitable rule no=2
0.354 0.378 0.463 0.354 0.378 0.401 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.028 0.000 0.000
***** data no:73 *****
Symptoms: 0.000 0.000 -0.210 0.076 0.142 0.120 -0.074 -2.985
Faulty group=1 Most suitable rule no=3
0.005 0.000 0.000 0.930 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:74 *****
Symptoms: 0.000 0.000 -0.142 -1.012 -1.237 -1.145 -0.232 -1.060
Faulty group=1 Most suitable rule no=0
0.588 0.047 0.353 0.353 0.047 0.412 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:75 *****
Symptoms: 0.000 0.000 -0.850 -1.332 -1.740 -1.434 0.057 -1.754
Faulty group=1 Most suitable rule no=2
0.415 0.283 0.522 0.420 0.283 0.415 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.019 0.000 0.000
***** data no:76 *****

Symptoms: 0.000 0.000 -2.194 -2.628 -3.186 -2.797 0.069 -3.274
Faulty group=1 Most suitable rule no=1
0.000 0.731 0.068 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:77 *****
Symptoms: 0.000 0.000 -0.158 -0.862 -1.479 -0.941 0.058 -1.356
Faulty group=1 Most suitable rule no=0
0.507 0.053 0.452 0.452 0.053 0.493 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.019 0.000 0.000
***** data no:78 *****
Symptoms: 0.000 0.000 -1.147 -1.171 -1.926 -1.476 0.042 -1.683
Faulty group=1 Most suitable rule no=2
0.358 0.382 0.508 0.358 0.382 0.439 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.014 0.000 0.000
***** data no:79 *****
Symptoms: 0.000 0.000 -2.237 -2.255 -3.168 -2.633 0.142 -3.314
Faulty group=1 Most suitable rule no=1
0.000 0.746 0.122 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:80 *****
Symptoms: 0.000 0.000 -1.463 -1.953 -2.510 -2.126 0.137 -2.682
Faulty group=1 Most suitable rule no=1
0.106 0.488 0.291 0.163 0.106 0.106 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.046 0.000 0.000
***** data no:81 *****
Symptoms: 0.000 0.000 -5.324 -7.161 -9.426 -7.633 -0.413 -9.298
Faulty group=1 Most suitable rule no=1
0.000 0.862 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:82 *****
Symptoms: 0.000 0.000 -8.641 -10.777 -13.463 -11.393 -0.011 -13.344
Faulty group=1 Most suitable rule no=1
0.000 0.996 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:83 *****
Symptoms: 0.000 0.000 -9.593 -10.467 -12.400 -11.062 0.494 -22.093
Faulty group=1 Most suitable rule no=1
0.000 0.835 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:84 *****
Symptoms: 0.000 0.000 -15.659 -18.265 -22.371 -19.452 -0.025 -22.459
Faulty group=1 Most suitable rule no=1
0.000 0.992 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:85 *****
Symptoms: 0.000 0.000 -7.610 -10.205 -13.600 -10.954 -0.037 -13.620
Faulty group=1 Most suitable rule no=1
0.000 0.988 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:86 *****
Symptoms: 0.000 0.000 -4.114 -6.813 -9.605 -7.387 -0.232 -9.454
Faulty group=1 Most suitable rule no=1
0.000 0.923 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:87 *****
Symptoms: 0.000 0.000 -3.312 -6.160 -10.353 -6.816 0.058 -10.243
Faulty group=1 Most suitable rule no=1
0.000 0.981 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:88 *****
Symptoms: 0.000 0.000 -6.687 -9.441 -14.397 -10.550 0.042 -14.189
Faulty group=1 Most suitable rule no=1
0.000 0.986 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:89 *****
Symptoms: 0.000 0.000 -13.897 -17.215 -23.177 -18.764 0.047 -23.297
Faulty group=1 Most suitable rule no=1
0.000 0.984 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:90 *****
Symptoms: 0.000 0.000 -9.909 -13.437 -18.734 -14.643 0.042 -18.889
Faulty group=1 Most suitable rule no=1
0.000 0.986 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:91 *****
Symptoms: 0.000 0.000 -0.880 -0.797 -1.225 -0.616 0.166 -0.161
Faulty group=2 Most suitable rule no=0
0.592 0.054 0.054 0.054 0.205 0.408 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.055 0.000 0.000
***** data no:92 *****
Symptoms: 0.000 0.000 -1.154 -1.337 -1.709 -1.381 0.039 0.002
Faulty group=2 Most suitable rule no=5
0.430 0.000 0.000 0.000 0.385 0.540 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.013 0.000 0.000
***** data no:93 *****
Symptoms: 0.000 0.000 -1.555 -1.684 -2.017 -1.883 0.123 0.190
Faulty group=2 Most suitable rule no=4
0.328 0.000 0.000 0.000 0.518 0.372 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.041 0.000 0.000
***** data no:94 *****
Symptoms: 0.000 0.000 -0.322 -0.674 -1.065 -0.823 0.084 0.001
Faulty group=2 Most suitable rule no=0
0.645 0.000 0.000 0.000 0.107 0.355 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.028 0.000 0.000
***** data no:95 *****
Symptoms: 0.000 0.000 -0.837 -0.946 -1.210 -0.968 0.082 -0.008
Faulty group=2 Most suitable rule no=0
0.597 0.003 0.003 0.003 0.279 0.403 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.027 0.000 0.000
***** data no:96 *****
Symptoms: 0.000 0.000 -2.402 -2.751 -3.437 -2.942 -0.147 -0.026
Faulty group=2 Most suitable rule no=4
0.000 0.009 0.009 0.000 0.801 0.019 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:97 *****
Symptoms: 0.000 0.000 -0.750 -0.785 -0.817 -0.318 -0.162 -0.031
Faulty group=2 Most suitable rule no=0
0.728 0.010 0.010 0.010 0.106 0.272 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:98 *****
Symptoms: 0.000 0.000 -0.921 -1.085 -1.455 -0.992 0.055 -0.078
Faulty group=2 Most suitable rule no=0
0.515 0.026 0.026 0.026 0.307 0.485 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.018 0.000 0.000
***** data no:99 *****
Symptoms: 0.000 0.000 -2.014 -2.383 -3.016 -2.438 0.231 0.099
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.671 0.187 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.077 0.000 0.000
***** data no:100 *****
Symptoms: 0.000 0.000 -1.228 -1.767 -2.406 -1.868 -0.249 -0.010
Faulty group=2 Most suitable rule no=4
0.198 0.003 0.003 0.003 0.409 0.377 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:101 *****
Symptoms: 0.000 0.000 -0.724 -0.992 -1.174 -1.041 -0.319 0.011
Faulty group=2 Most suitable rule no=0

0.609 0.000 0.000 0.000 0.241 0.391 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:102 *****
Symptoms: 0.000 0.000 -1.134 -1.597 -1.932 -1.610 0.083 -0.026
Faulty group=2 Most suitable rule no=5
0.356 0.009 0.009 0.009 0.378 0.463 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.028 0.000 0.000
***** data no:103 *****
Symptoms: 0.000 0.000 -0.210 0.076 0.139 0.120 0.020 -0.031
Faulty group=2 Most suitable rule no=0
0.930 0.000 0.000 0.010 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:104 *****
Symptoms: 0.000 0.000 -0.010 -0.888 -0.978 -0.957 -0.232 0.152
Faulty group=2 Most suitable rule no=0
0.674 0.000 0.000 0.000 0.003 0.326 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:105 *****
Symptoms: 0.000 0.000 -0.735 -1.130 -1.525 -1.256 0.057 0.050
Faulty group=2 Most suitable rule no=5
0.492 0.000 0.000 0.000 0.245 0.508 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.019 0.000 0.000
***** data no:106 *****
Symptoms: 0.000 0.000 -2.301 -2.739 -3.336 -2.926 0.069 -0.020
Faulty group=2 Most suitable rule no=4
0.000 0.007 0.007 0.000 0.767 0.025 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.023 0.000 0.000
***** data no:107 *****
Symptoms: 0.000 0.000 -0.036 -0.621 -1.076 -0.690 0.058 -0.066
Faulty group=2 Most suitable rule no=0
0.641 0.012 0.022 0.022 0.012 0.359 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.012 0.000 0.000
***** data no:108 *****
Symptoms: 0.000 0.000 -0.945 -0.835 -1.471 -1.157 0.042 0.237
Faulty group=2 Most suitable rule no=0
0.510 0.000 0.000 0.000 0.278 0.490 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.014 0.000 0.000
***** data no:109 *****
Symptoms: 0.000 0.000 -1.994 -1.950 -2.752 -2.298 0.142 0.080
Faulty group=2 Most suitable rule no=4
0.083 0.000 0.000 0.000 0.650 0.234 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.047 0.000 0.000
***** data no:110 *****
Symptoms: 0.000 0.000 -1.226 -1.607 -2.047 -1.769 0.137 -0.092
Faulty group=2 Most suitable rule no=5
0.318 0.031 0.031 0.031 0.409 0.410 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.046 0.000 0.000
***** data no:111 *****
Symptoms: 0.000 0.000 -6.573 -8.816 -11.672 -9.441 -0.413 0.047
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.862 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:112 *****
Symptoms: 0.000 0.000 -11.359 -14.119 -17.644 -14.934 -0.105 0.046
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.965 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:113 *****
Symptoms: 0.000 0.000 -16.356 -18.125 -21.497 -19.153 0.400 0.162
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.867 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:114 *****

Symptoms: 0.000 0.000 -19.842 -23.145 -28.359 -24.633 -0.120 0.220
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.927 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:115 *****
Symptoms: 0.000 0.000 -9.244 -12.323 -16.479 -13.258 -0.037 0.130
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.957 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:116 *****
Symptoms: 0.000 0.000 -4.789 -7.800 -11.028 -8.454 -0.326 0.184
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.891 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:117 *****
Symptoms: 0.000 0.000 -3.483 -6.481 -10.823 -7.150 0.058 -0.018
Faulty group=2 Most suitable rule no=4
0.000 0.006 0.000 0.000 0.981 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:118 *****
Symptoms: 0.000 0.000 -7.314 -10.382 -15.802 -11.566 -0.053 0.309
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.897 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:119 *****
Symptoms: 0.000 0.000 -15.954 -19.861 -26.715 -21.608 0.047 0.297
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.901 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:120 *****
Symptoms: 0.000 0.000 -11.198 -15.226 -21.213 -16.554 0.042 0.027
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.986 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:121 *****
Symptoms: 0.000 0.000 -3.258 -4.109 -5.736 -4.219 -0.211 -0.148
Faulty group=3 Most suitable rule no=4
0.000 0.049 0.000 0.000 0.930 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:122 *****
Symptoms: 0.000 0.000 -4.182 -5.150 -6.587 -5.496 -0.432 0.028
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.856 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:123 *****
Symptoms: 0.000 0.000 -5.293 -5.876 -7.032 -6.395 -0.537 0.243
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.821 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:124 *****
Symptoms: 0.000 0.000 -2.770 -4.162 -6.029 -4.668 -0.293 0.016
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.902 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:125 *****
Symptoms: 0.000 0.000 -3.856 -4.931 -6.541 -5.309 -0.389 0.022
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.870 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:126 *****
Symptoms: 0.000 0.000 -6.494 -7.454 -9.118 -7.982 -0.805 0.034
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.732 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:127 *****
Symptoms: 0.000 0.000 -3.586 -5.073 -7.268 -5.075 -0.538 -0.011
Faulty group=3 Most suitable rule no=4
0.000 0.004 0.000 0.000 0.821 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:128 *****
Symptoms: 0.000 0.000 -4.352 -5.885 -8.305 -6.277 -0.322 -0.038
Faulty group=3 Most suitable rule no=4
0.000 0.013 0.000 0.000 0.893 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:129 *****
Symptoms: 0.000 0.000 -6.899 -8.304 -10.647 -8.881 -0.335 0.179
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.888 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:130 *****
Symptoms: 0.000 0.000 -5.470 -7.222 -9.715 -7.822 -0.718 0.030
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.761 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:131 *****
Symptoms: 0.000 0.000 -6.030 -7.910 -11.013 -8.425 -0.683 -0.121
Faulty group=3 Most suitable rule no=4
0.000 0.040 0.000 0.000 0.772 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:132 *****
Symptoms: 0.000 0.000 -8.026 -9.972 -12.785 -10.704 -0.997 0.055
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.668 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:133 *****
Symptoms: 0.000 0.000 -11.269 -12.615 -15.073 -13.585 -1.197 0.324
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.601 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:134 *****
Symptoms: 0.000 0.000 -5.565 -8.182 -11.688 -9.022 -0.764 0.031
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.745 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:135 *****
Symptoms: 0.000 0.000 -7.465 -9.722 -12.943 -10.488 -0.860 0.052
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.713 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:136 *****
Symptoms: 0.000 0.000 -12.522 -14.368 -17.513 -15.385 -1.369 0.110
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.544 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:137 *****
Symptoms: 0.000 0.000 -6.694 -9.762 -14.387 -10.311 -0.914 0.009
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.695 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:138 *****
Symptoms: 0.000 0.000 -8.243 -11.338 -16.108 -12.292 -0.793 0.002
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.736 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:139 *****
Symptoms: 0.000 0.000 -13.133 -15.905 -20.399 -17.080 -0.901 0.260
Faulty group=3 Most suitable rule no=4

0.000 0.000 0.000 0.000 0.700 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:140 *****
Symptoms: 0.000 0.000 -10.550 -13.788 -18.496 -14.934 -1.188 0.110
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.604 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:141 *****
Symptoms: 0.000 0.000 -3.039 -4.102 -5.317 -4.418 -0.788 0.035
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.737 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:142 *****
Symptoms: 0.000 0.000 -4.016 -5.100 -6.355 -5.414 -0.482 -0.002
Faulty group=3 Most suitable rule no=4
0.000 0.001 0.000 0.000 0.839 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:143 *****
Symptoms: 0.000 0.000 -0.210 0.076 0.142 0.103 -0.264 -0.031
Faulty group=3 Most suitable rule no=0
0.912 0.000 0.000 0.010 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:144 *****
Symptoms: 0.000 0.000 -2.502 -4.468 -6.195 -4.909 -0.608 0.168
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.797 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:145 *****
Symptoms: 0.000 0.000 -3.992 -5.416 -7.242 -5.899 -0.414 0.082
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.862 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:146 *****
Symptoms: 0.000 0.000 -6.590 -7.730 -9.437 -8.274 -0.591 0.028
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.803 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:147 *****
Symptoms: 0.000 0.000 -2.554 -4.876 -8.182 -5.457 -0.224 -0.042
Faulty group=3 Most suitable rule no=4
0.000 0.014 0.000 0.000 0.851 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:148 *****
Symptoms: 0.000 0.000 -4.443 -6.079 -9.364 -6.953 -0.335 0.261
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.888 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:149 *****
Symptoms: 0.000 0.000 -7.238 -8.666 -11.741 -9.598 -0.330 0.152
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.890 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:150 *****
Symptoms: 0.000 0.000 -5.669 -7.667 -10.566 -8.402 -0.335 -0.044
Faulty group=3 Most suitable rule no=4
0.000 0.015 0.000 0.000 0.888 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:151 *****
Symptoms: 0.000 0.000 -5.799 -7.813 -10.285 -8.436 -1.258 0.047
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.581 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:152 *****

Symptoms: 0.000 0.000 -7.814 -9.771 -12.197 -10.411 -0.953 0.022
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.682 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:153 *****
Symptoms: 0.000 0.000 -4.262 -4.474 -5.262 -4.769 -0.738 0.017
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.754 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:154 *****
Symptoms: 0.000 0.000 -5.287 -8.541 -12.084 -9.363 -0.984 0.200
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.672 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:155 *****
Symptoms: 0.000 0.000 -7.888 -10.558 -14.105 -11.455 -0.885 0.114
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.705 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:156 *****
Symptoms: 0.000 0.000 -12.682 -14.791 -18.116 -15.848 -1.156 0.124
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.615 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:157 *****
Symptoms: 0.000 0.000 -5.219 -9.451 -15.743 -10.537 -0.601 -0.018
Faulty group=3 Most suitable rule no=4
0.000 0.006 0.000 0.000 0.800 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:158 *****
Symptoms: 0.000 0.000 -8.305 -11.861 -18.046 -13.311 -0.712 0.309
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.763 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:159 *****
Symptoms: 0.000 0.000 -13.496 -16.706 -22.496 -18.302 -0.896 0.249
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.701 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:160 *****
Symptoms: 0.000 0.000 -10.740 -14.592 -20.311 -15.955 -0.806 0.027
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.731 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:161 *****
Symptoms: 0.000 0.000 4.844 27.058 -0.280 7.918 0.166 -0.188
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.907 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:162 *****
Symptoms: 0.000 0.000 5.304 21.806 -0.110 7.280 0.039 -0.052
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.963 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:163 *****
Symptoms: 0.000 0.000 0.541 10.730 -6.475 0.584 -0.537 0.163
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.805 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:164 *****
Symptoms: 0.000 0.000 5.644 29.538 -0.160 8.363 0.084 -0.044
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.947 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:165 *****
Symptoms: 0.000 0.000 5.703 25.330 0.146 8.322 0.082 -0.053
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.951 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.049
***** data no:166 *****
Symptoms: 0.000 0.000 5.767 17.463 -0.128 7.030 -0.147 -0.131
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.951 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:167 *****
Symptoms: 0.000 0.000 5.976 33.381 0.174 10.848 -0.162 -0.090
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.942 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.058
***** data no:168 *****
Symptoms: 0.000 0.000 6.401 29.556 -0.090 10.176 0.055 -0.137
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.954 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:169 *****
Symptoms: 0.000 0.000 6.756 22.010 -0.106 9.037 0.231 -0.021
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.923 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:170 *****
Symptoms: 0.000 0.000 6.837 25.000 -0.353 9.353 -0.249 -0.110
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.882 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:171 *****
Symptoms: 0.000 0.000 4.805 24.788 -0.188 6.895 -0.319 -0.013
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.894 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:172 *****
Symptoms: 0.000 0.000 5.107 19.863 -0.262 6.516 0.083 -0.074
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.913 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:173 *****
Symptoms: 0.000 0.000 -0.149 10.106 -6.977 0.142 -0.833 -0.031
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.722 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:174 *****
Symptoms: 0.000 0.000 6.087 30.647 -0.092 8.642 -0.232 0.120
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.923 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:175 *****
Symptoms: 0.000 0.000 6.141 25.188 -0.041 8.430 0.057 -0.013
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.981 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:176 *****
Symptoms: 0.000 0.000 6.005 18.369 -0.124 7.352 0.069 -0.132
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.956 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:177 *****
Symptoms: 0.000 0.000 6.614 39.917 -0.285 11.687 0.058 -0.114
Faulty group=4 Most suitable rule no=6

0.000 0.000 0.000 0.000 0.000 0.000 0.905 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:178 *****
Symptoms: 0.000 0.000 6.678 33.654 -0.177 11.331 0.042 0.141
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.941 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:179 *****
Symptoms: 0.000 0.000 7.165 25.526 -0.046 10.355 0.142 -0.064
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.953 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:180 *****
Symptoms: 0.000 0.000 7.237 28.632 -0.121 10.747 0.137 -0.212
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.929 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:181 *****
Symptoms: 0.000 0.000 10.039 54.152 -0.340 15.671 0.166 -0.227
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.887 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:182 *****
Symptoms: 0.000 0.000 10.719 43.677 -0.199 14.531 0.039 -0.091
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.934 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:183 *****
Symptoms: 0.000 0.000 0.581 20.823 -13.666 0.600 -1.102 0.163
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.633 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:184 *****
Symptoms: 0.000 0.000 11.141 59.040 -0.227 16.816 0.084 -0.074
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.924 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:185 *****
Symptoms: 0.000 0.000 11.429 50.447 0.051 16.452 0.082 -0.098
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.967 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.017
***** data no:186 *****
Symptoms: 0.000 0.000 8.376 31.431 -4.581 10.202 -0.429 -0.176
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:187 *****
Symptoms: 0.000 0.000 12.229 66.810 0.073 21.221 -0.162 -0.150
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.946 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.024
***** data no:188 *****
Symptoms: 0.000 0.000 12.960 59.070 -0.216 20.201 0.055 -0.217
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.928 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:189 *****
Symptoms: 0.000 0.000 13.534 43.929 -0.304 17.914 0.231 -0.122
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.899 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:190 *****

Symptoms: 0.000 0.000 13.619 50.152 -0.508 18.778 -0.249 -0.190
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.831 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:191 *****
Symptoms: 0.000 0.000 9.763 49.866 -0.248 14.002 -0.319 -0.048
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.894 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:192 *****
Symptoms: 0.000 0.000 10.209 39.914 -0.338 13.143 0.083 -0.122
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.887 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:193 *****
Symptoms: 0.000 0.000 -0.108 20.136 -14.118 0.159 -1.497 -0.031
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.501 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:194 *****
Symptoms: 0.000 0.000 11.730 61.442 -0.163 17.503 -0.232 0.073
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.923 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:195 *****
Symptoms: 0.000 0.000 12.111 50.297 -0.143 16.847 0.057 -0.061
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.952 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:196 *****
Symptoms: 0.000 0.000 11.943 36.705 -0.283 14.689 0.069 -0.212
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.906 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:197 *****
Symptoms: 0.000 0.000 12.970 80.054 -0.406 23.521 0.058 -0.185
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.865 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:198 *****
Symptoms: 0.000 0.000 13.694 67.202 -0.329 22.788 0.042 0.069
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.890 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:199 *****
Symptoms: 0.000 0.000 14.624 50.764 -0.271 20.645 0.142 -0.185
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.910 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:200 *****
Symptoms: 0.000 0.000 14.581 57.371 -0.308 21.617 0.137 -0.308
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.897 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:201 *****
Symptoms: 0.000 0.000 -99.000 -50.000 -50.000 -50.000 -50.000 -99.000
Faulty group=7 Most suitable rule no=26
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000

Appendix E

Fuzzy diagnostic outputs using product operation

***** data no:1 *****
Symptom: 2.000000 0.000000 3.149510 1.621960 1.946690 1.593990 0.202480 2.115000
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.009 0.000 0.000 0.082
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.017
***** data no:2 *****
Symptom: 2.000000 0.000000 3.059900 1.469110 1.737790 1.466450 0.272710 2.069180
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.009 0.000 0.000 0.058
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.013
***** data no:3 *****
Symptom: 2.000000 0.000000 3.102190 1.564910 2.022100 1.651520 0.088430 2.095220
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.009 0.000 0.000 0.087
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.019
***** data no:4 *****
Symptom: 2.000000 0.000000 3.125020 1.495310 1.871930 1.609010 0.128580 2.051120
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.010 0.000 0.000 0.073
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.017
***** data no:5 *****
Symptom: 2.000000 0.000000 2.993830 1.634560 2.274090 1.838030 -0.033730 2.034690
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.009 0.000 0.000 0.113
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.027
***** data no:6 *****
Symptom: 2.000000 0.000000 3.041080 1.678780 2.173420 1.690500 0.216020 2.061600
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.008 0.000 0.000 0.097
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.022
***** data no:7 *****
Symptom: 2.000000 0.000000 3.032400 1.615260 2.010460 1.399510 0.147740 2.032870
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.008 0.000 0.000 0.072
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.017
***** data no:8 *****
Symptom: 2.000000 0.000000 3.256270 1.466740 1.920910 1.506110 0.096760 2.126960
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.008 0.000 0.000 0.072
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.015
***** data no:9 *****
Symptom: 2.000000 0.000000 3.334860 1.423740 1.892360 1.498530 -0.136200 2.093140
Faulty group=5 Most suitable rule no=9
0.000 0.000 0.000 0.000 0.000 0.000 0.008 0.000 0.000 0.066
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.014
***** data no:10 *****
Symptom: 2.000000 0.000000 2.728150 1.346400 2.002690 1.730230 0.029970 1.925690
Faulty group=5 Most suitable rule no=9
0.001 0.000 0.000 0.000 0.000 0.000 0.009 0.000 0.000 0.067
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.019
***** data no:11 *****
Symptom: -2.000000 0.000000 -3.054220 -1.403290 -1.741580 -1.415020 -0.065610 -2.029490
Faulty group=5 Most suitable rule no=8
0.000 0.028 0.000 0.000 0.014 0.000 0.000 0.000 0.057 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:12 *****
Symptom: -2.000000 0.000000 -3.123790 -1.450480 -1.813280 -1.461370 -0.074990 -2.081210
Faulty group=5 Most suitable rule no=8
0.000 0.032 0.000 0.000 0.014 0.000 0.000 0.000 0.064 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:13 *****
Symptom: -2.000000 0.000000 -3.146810 -1.544690 -1.921990 -1.672130 0.193400 -2.027230
Faulty group=5 Most suitable rule no=8
0.000 0.039 0.000 0.000 0.019 0.000 0.000 0.000 0.077 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:14 *****
Symptom: -2.000000 0.000000 -3.025330 -1.492030 -1.817960 -1.610200 0.208470 -2.046980
Faulty group=5 Most suitable rule no=8
0.000 0.034 0.000 0.000 0.016 0.000 0.000 0.000 0.068 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:15 *****
Symptom: -2.000000 0.000000 -2.909330 -1.484820 -1.862880 -1.562500 -0.082970 -2.004900
Faulty group=5 Most suitable rule no=8
0.000 0.034 0.001 0.001 0.017 0.000 0.000 0.000 0.067 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:16 *****
Symptom: -2.000000 0.000000 -2.703050 -1.312490 -1.854070 -1.503220 0.164320 -1.912380
Faulty group=5 Most suitable rule no=8
0.001 0.025 0.003 0.002 0.014 0.002 0.000 0.000 0.049 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:17 *****
Symptom: -2.000000 0.000000 -2.558680 -1.210240 -1.949410 -1.677220 0.190560 -1.932280
Faulty group=5 Most suitable rule no=8
0.002 0.025 0.005 0.003 0.014 0.003 0.000 0.000 0.050 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:18 *****
Symptom: -2.000000 0.000000 -2.557930 -1.340130 -2.004430 -1.642360 0.072700 -1.915040
Faulty group=5 Most suitable rule no=8
0.001 0.029 0.005 0.003 0.016 0.003 0.000 0.000 0.058 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:19 *****
Symptom: -2.000000 0.000000 -2.567400 -1.578360 -2.148950 -1.661230 -0.032730 -1.908410
Faulty group=5 Most suitable rule no=8
0.001 0.037 0.005 0.002 0.021 0.003 0.000 0.000 0.075 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:20 *****
Symptom: -2.000000 0.000000 -2.725740 -1.571680 -2.066080 -1.765600 -0.112750 -1.903880
Faulty group=5 Most suitable rule no=8
0.001 0.039 0.003 0.001 0.023 0.001 0.000 0.000 0.079 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:21 *****
Symptom: 0.000000 2.000000 -1.976880 -1.584510 -2.263070 -2.761180 0.334690 0.080940
Faulty group=6 Most suitable rule no=11
0.001 0.000 0.000 0.000 0.070 0.003 0.000 0.000 0.000 0.000
0.000 0.139 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:22 *****
Symptom: 0.000000 2.000000 -2.081970 -1.768770 -2.300190 -2.834510 0.258540 0.022730
Faulty group=6 Most suitable rule no=11
0.000 0.000 0.000 0.000 0.090 0.002 0.000 0.000 0.000 0.000
0.000 0.179 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:23 *****
Symptom: 0.000000 2.000000 -2.002530 -1.641850 -2.156830 -2.700380 0.177080 0.049140
Faulty group=6 Most suitable rule no=11
0.001 0.000 0.000 0.000 0.073 0.003 0.000 0.000 0.000 0.000
0.000 0.146 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:24 *****
Symptom: 0.000000 2.000000 -2.033040 -1.719140 -2.160700 -2.720020 0.138330 -0.004900
Faulty group=6 Most suitable rule no=11
0.001 0.000 0.000 0.000 0.081 0.003 0.000 0.000 0.000 0.000
0.000 0.161 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:25 *****
Symptom: 0.000000 2.000000 -1.962820 -1.506530 -1.883810 -2.469810 0.005670 0.027370
Faulty group=6 Most suitable rule no=11

0.004 0.000 0.000 0.000 0.056 0.006 0.000 0.000 0.000 0.000
0.000 0.112 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:26 *****
Symptom: 0.000000 2.000000 -1.713900 -1.300350 -1.769810 -2.270710 0.158140 0.014930
Faulty group=6 Most suitable rule no=11
0.008 0.000 0.000 0.000 0.035 0.011 0.000 0.000 0.000 0.000
0.000 0.069 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:27 *****
Symptom: 0.000000 2.000000 -1.530980 -1.204730 -1.966140 -2.364010 0.220550 -0.038540
Faulty group=6 Most suitable rule no=11
0.007 0.000 0.000 0.000 0.032 0.012 0.000 0.000 0.000 0.000
0.000 0.065 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:28 *****
Symptom: 0.000000 2.000000 -1.453010 -1.499530 -2.074720 -2.594940 0.064350 0.105040
Faulty group=6 Most suitable rule no=11
0.003 0.000 0.000 0.000 0.046 0.008 0.000 0.000 0.000 0.000
0.000 0.091 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:29 *****
Symptom: 0.000000 2.000000 -1.722630 -1.577360 -2.029850 -2.763800 -0.107160 0.053400
Faulty group=6 Most suitable rule no=11
0.002 0.000 0.000 0.000 0.059 0.003 0.000 0.000 0.000 0.000
0.000 0.119 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:30 *****
Symptom: 0.000000 2.000000 -1.742420 -1.553640 -2.073620 -2.431810 -0.027040 -0.072150
Faulty group=6 Most suitable rule no=11
0.004 0.001 0.000 0.000 0.054 0.009 0.000 0.000 0.000 0.000
0.000 0.109 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:31 *****
Symptom: 0.000000 -2.000000 1.854800 1.584800 1.841890 2.410220 0.252250 0.078590
Faulty group=6 Most suitable rule no=10
0.004 0.000 0.000 0.000 0.000 0.000 0.030 0.000 0.000 0.000
0.096 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.048
***** data no:32 *****
Symptom: 0.000000 -2.000000 1.766270 1.409830 1.620200 2.273920 0.336840 0.040290
Faulty group=6 Most suitable rule no=10
0.007 0.000 0.000 0.000 0.000 0.000 0.028 0.000 0.000 0.000
0.066 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.033
***** data no:33 *****
Symptom: 0.000000 -2.000000 1.839770 1.503860 1.905850 2.473290 0.140230 0.050860
Faulty group=6 Most suitable rule no=10
0.004 0.000 0.000 0.000 0.000 0.000 0.029 0.000 0.000 0.000
0.101 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.050
***** data no:34 *****
Symptom: 0.000000 -2.000000 1.862050 1.414020 1.738280 2.446260 0.176730 -0.004430
Faulty group=6 Most suitable rule no=10
0.005 0.000 0.000 0.000 0.000 0.000 0.031 0.000 0.000 0.000
0.087 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.043
***** data no:35 *****
Symptom: 0.000000 -2.000000 1.754840 1.566110 2.187490 2.667950 -0.014840 0.013850
Faulty group=6 Most suitable rule no=10
0.002 0.000 0.000 0.000 0.000 0.000 0.024 0.000 0.000 0.000
0.131 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.065
***** data no:36 *****
Symptom: 0.000000 -2.000000 1.664640 1.645840 2.127480 2.572970 0.210240 -0.022980
Faulty group=6 Most suitable rule no=10
0.003 0.000 0.000 0.000 0.000 0.000 0.023 0.000 0.000 0.000
0.114 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.057
***** data no:37 *****
Symptom: 0.000000 -2.000000 0.056530 0.190360 -0.022800 -0.055280 0.199750 -0.049820
Faulty group=6 Most suitable rule no=0
0.274 0.000 0.000 0.005 0.000 0.002 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:38 *****

Symptom: 0.000000 -2.000000 2.027530 1.447860 1.880070 2.394850 0.130670 0.077090
Faulty group=6 Most suitable rule no=10
0.004 0.000 0.000 0.000 0.000 0.000 0.030 0.000 0.000 0.000
0.101 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.051
***** data no:39 *****
Symptom: 0.000000 -2.000000 2.153670 1.379420 1.844490 2.484850 -0.114210 0.025280
Faulty group=6 Most suitable rule no=10
0.003 0.000 0.000 0.000 0.000 0.000 0.033 0.000 0.000 0.000
0.107 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.053
***** data no:40 *****
Symptom: 0.000000 -2.000000 1.585680 1.308150 1.916180 2.626760 0.032410 -0.106190
Faulty group=6 Most suitable rule no=10
0.004 0.000 0.000 0.000 0.000 0.000 0.023 0.000 0.000 0.000
0.082 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.041
***** data no:41 *****
Symptom: 0.000000 0.000000 -0.050020 0.011030 -0.218660 -0.179170 0.304720 0.081680
Faulty group=0 Most suitable rule no=0
0.746 0.000 0.000 0.000 0.000 0.059 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:42 *****
Symptom: 0.000000 0.000000 -0.160820 -0.181440 -0.358290 -0.296470 0.310910 0.032470
Faulty group=0 Most suitable rule no=0
0.626 0.000 0.000 0.000 0.000 0.085 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:43 *****
Symptom: 0.000000 0.000000 -0.072700 -0.061710 -0.135260 -0.119430 0.163800 0.050830
Faulty group=0 Most suitable rule no=0
0.814 0.000 0.000 0.000 0.000 0.038 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:44 *****
Symptom: 0.000000 0.000000 -0.081350 -0.154100 -0.222720 -0.143360 0.163190 -0.006180
Faulty group=0 Most suitable rule no=0
0.768 0.000 0.000 0.002 0.000 0.062 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:45 *****
Symptom: 0.000000 0.000000 -0.087590 0.038880 0.152860 0.101000 -0.005990 0.021090
Faulty group=0 Most suitable rule no=0
0.871 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:46 *****
Symptom: 0.000000 0.000000 -0.009560 0.181930 0.182040 0.159170 0.188010 -0.002510
Faulty group=0 Most suitable rule no=0
0.780 0.000 0.000 0.001 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:47 *****
Symptom: 0.000000 0.000000 0.056530 0.190360 -0.022800 -0.055280 0.199750 -0.049820
Faulty group=0 Most suitable rule no=0
0.822 0.000 0.000 0.014 0.000 0.006 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:48 *****
Symptom: 0.000000 0.000000 0.304760 -0.019240 -0.098000 -0.093570 0.101770 0.091770
Faulty group=0 Most suitable rule no=0
0.783 0.000 0.000 0.000 0.000 0.026 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:49 *****
Symptom: 0.000000 0.000000 0.227780 -0.094870 -0.096290 -0.141570 -0.118920 0.038300
Faulty group=0 Most suitable rule no=0
0.782 0.000 0.000 0.000 0.000 0.026 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:50 *****
Symptom: 0.000000 0.000000 -0.060820 -0.116100 -0.080120 0.102030 0.007000 -0.090490
Faulty group=0 Most suitable rule no=0
0.857 0.000 0.001 0.027 0.000 0.024 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:51 *****
Symptom: 0.000000 0.000000 -0.067670 0.087890 0.209650 0.165290 -0.043280 -0.047860
Faulty group=0 Most suitable rule no=0
0.809 0.000 0.000 0.013 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:52 *****
Symptom: 0.000000 0.000000 -0.103730 0.067050 0.155810 0.143370 -0.088110 -0.107770
Faulty group=0 Most suitable rule no=0
0.797 0.000 0.000 0.030 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:53 *****
Symptom: 0.000000 0.000000 -0.034550 0.041790 0.084480 0.033070 0.219420 -0.089570
Faulty group=0 Most suitable rule no=0
0.842 0.000 0.000 0.026 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:54 *****
Symptom: 0.000000 0.000000 0.077550 0.049750 0.144510 0.026000 0.180630 -0.066240
Faulty group=0 Most suitable rule no=0
0.831 0.000 0.000 0.019 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:55 *****
Symptom: 0.000000 0.000000 0.101710 -0.015360 0.053680 -0.005510 -0.071890 -0.007910
Faulty group=0 Most suitable rule no=0
0.917 0.000 0.000 0.002 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:56 *****
Symptom: 0.000000 0.000000 0.202950 0.128680 0.055290 -0.007970 0.184320 0.071560
Faulty group=0 Most suitable rule no=0
0.800 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:57 *****
Symptom: 0.000000 0.000000 0.245900 0.170280 -0.006090 -0.249050 0.144010 0.053330
Faulty group=0 Most suitable rule no=0
0.741 0.000 0.000 0.000 0.000 0.002 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:58 *****
Symptom: 0.000000 0.000000 0.084500 0.049760 -0.016990 -0.093110 0.045220 0.025330
Faulty group=0 Most suitable rule no=0
0.899 0.000 0.000 0.000 0.000 0.005 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:59 *****
Symptom: 0.000000 0.000000 0.220820 -0.133390 -0.190490 -0.077480 -0.031940 0.032330
Faulty group=0 Most suitable rule no=0
0.790 0.000 0.000 0.000 0.000 0.054 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:60 *****
Symptom: 0.000000 0.000000 0.211160 -0.086680 -0.145260 -0.156240 -0.157520 0.055510
Faulty group=0 Most suitable rule no=0
0.757 0.000 0.000 0.000 0.000 0.039 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:61 *****
Symptom: 0.000000 0.000000 -0.981560 -0.905910 -1.411540 -0.767110 0.165980 -1.354140
Faulty group=1 Most suitable rule no=0
0.096 0.005 0.070 0.079 0.006 0.085 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:62 *****
Symptom: 0.000000 0.000000 -1.207090 -1.380930 -1.793580 -1.448280 0.038800 -1.780580
Faulty group=1 Most suitable rule no=2
0.027 0.031 0.058 0.039 0.021 0.040 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:63 *****
Symptom: 0.000000 0.000000 -1.202240 -1.264910 -1.547630 -1.461140 0.123350 -3.035560
Faulty group=1 Most suitable rule no=2

0.000 0.041 0.088 0.083 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:64 *****
Symptom: 0.000000 0.000000 -0.442500 -0.851040 -1.299810 -1.002770 0.084250 -1.209150
Faulty group=1 Most suitable rule no=0
0.134 0.002 0.069 0.090 0.004 0.102 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:65 *****
Symptom: 0.000000 0.000000 -0.948050 -1.097540 -1.406040 -1.128670 0.082230 -1.653270
Faulty group=1 Most suitable rule no=3
0.063 0.011 0.068 0.077 0.009 0.055 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:66 *****
Symptom: 0.000000 0.000000 -2.216920 -2.540910 -3.181400 -2.722430 -0.147040 -3.270940
Faulty group=1 Most suitable rule no=1
0.000 0.540 0.004 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:67 *****
Symptom: 0.000000 0.000000 -0.913780 -1.053060 -1.160960 -0.565300 -0.162150 -1.484920
Faulty group=1 Most suitable rule no=0
0.107 0.004 0.066 0.105 0.004 0.068 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:68 *****
Symptom: 0.000000 0.000000 -1.096070 -1.321810 -1.817240 -1.257520 0.054680 -1.932060
Faulty group=1 Most suitable rule no=2
0.028 0.026 0.079 0.051 0.014 0.044 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:69 *****
Symptom: 0.000000 0.000000 -2.133220 -2.515460 -3.197830 -2.594010 0.231070 -3.211430
Faulty group=1 Most suitable rule no=1
0.000 0.476 0.006 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:70 *****
Symptom: 0.000000 0.000000 -1.411630 -2.019820 -2.710160 -2.117300 -0.248650 -2.549160
Faulty group=1 Most suitable rule no=1
0.001 0.157 0.036 0.004 0.028 0.006 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:71 *****
Symptom: 0.000000 0.000000 -0.801920 -1.092280 -1.314420 -1.153540 -0.318990 -1.171700
Faulty group=1 Most suitable rule no=0
0.088 0.006 0.044 0.056 0.009 0.068 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:72 *****
Symptom: 0.000000 0.000000 -1.133750 -1.597240 -1.936720 -1.609930 0.083350 -1.796780
Faulty group=1 Most suitable rule no=2
0.019 0.041 0.051 0.028 0.027 0.034 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:73 *****
Symptom: 0.000000 0.000000 -0.210020 0.076260 0.142330 0.119920 -0.074480 -2.985030
Faulty group=1 Most suitable rule no=3
0.004 0.000 0.000 0.804 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:74 *****
Symptom: 0.000000 0.000000 -0.142260 -1.011880 -1.237120 -1.145090 -0.232290 -1.060460
Faulty group=1 Most suitable rule no=0
0.137 0.001 0.052 0.075 0.002 0.096 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:75 *****
Symptom: 0.000000 0.000000 -0.850480 -1.331670 -1.740390 -1.434410 0.056840 -1.754110
Faulty group=1 Most suitable rule no=2
0.036 0.020 0.069 0.050 0.014 0.049 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:76 *****

Symptom: 0.000000 0.000000 -2.193660 -2.628360 -3.185910 -2.796890 0.068910 -3.273600
Faulty group=1 Most suitable rule no=1
0.000 0.584 0.002 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:77 *****
Symptom: 0.000000 0.000000 -0.158290 -0.861950 -1.478590 -0.941020 0.058370 -1.356080
Faulty group=1 Most suitable rule no=0
0.126 0.001 0.101 0.104 0.001 0.123 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:78 *****
Symptom: 0.000000 0.000000 -1.147200 -1.171290 -1.925900 -1.475540 0.041520 -1.683420
Faulty group=1 Most suitable rule no=2
0.030 0.026 0.068 0.038 0.020 0.053 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:79 *****
Symptom: 0.000000 0.000000 -2.236970 -2.255290 -3.167500 -2.632960 0.141650 -3.314360
Faulty group=1 Most suitable rule no=1
0.000 0.469 0.007 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:80 *****
Symptom: 0.000000 0.000000 -1.463000 -1.953430 -2.510490 -2.126210 0.136510 -2.681780
Faulty group=1 Most suitable rule no=1
0.001 0.161 0.037 0.007 0.019 0.004 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:81 *****
Symptom: 0.000000 0.000000 -5.324120 -7.161160 -9.426480 -7.632810 -0.412860 -9.297850
Faulty group=1 Most suitable rule no=1
0.000 0.862 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:82 *****
Symptom: 0.000000 0.000000 -8.640880 -10.777200 -13.462590 -11.392830 -0.010900 -13.343760
Faulty group=1 Most suitable rule no=1
0.000 0.996 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:83 *****
Symptom: 0.000000 0.000000 -9.592990 -10.466510 -12.399810 -11.062270 0.494350 -22.092590
Faulty group=1 Most suitable rule no=1
0.000 0.835 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:84 *****
Symptom: 0.000000 0.000000 -15.658920 -18.265490 -22.370899 -19.452190 -0.025320 -22.458599
Faulty group=1 Most suitable rule no=1
0.000 0.992 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:85 *****
Symptom: 0.000000 0.000000 -7.609620 -10.205260 -13.599780 -10.953620 -0.037380 -13.619630
Faulty group=1 Most suitable rule no=1
0.000 0.988 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:86 *****
Symptom: 0.000000 0.000000 -4.114300 -6.812920 -9.605340 -7.387230 -0.232290 -9.454370
Faulty group=1 Most suitable rule no=1
0.000 0.923 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:87 *****
Symptom: 0.000000 0.000000 -3.311970 -6.160020 -10.353430 -6.815990 0.058370 -10.243350
Faulty group=1 Most suitable rule no=1
0.000 0.981 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:88 *****
Symptom: 0.000000 0.000000 -6.687330 -9.440630 -14.397400 -10.549570 0.041520 -14.189030
Faulty group=1 Most suitable rule no=1
0.000 0.986 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:89 *****
Symptom: 0.000000 0.000000 -13.896520 -17.214661 -23.177370 -18.764191 0.047350 -23.296700
Faulty group=1 Most suitable rule no=1
0.000 0.984 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:90 *****
Symptom: 0.000000 0.000000 -9.909030 -13.437400 -18.734360 -14.642540 0.042220 -18.889500
Faulty group=1 Most suitable rule no=1
0.000 0.986 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:91 *****
Symptom: 0.000000 0.000000 -0.880140 -0.797310 -1.225470 -0.616420 0.165980 -0.161160
Faulty group=2 Most suitable rule no=0
0.218 0.000 0.009 0.012 0.006 0.151 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:92 *****
Symptom: 0.000000 0.000000 -1.154440 -1.337110 -1.709380 -1.381140 0.038800 0.001600
Faulty group=2 Most suitable rule no=5
0.078 0.000 0.000 0.000 0.044 0.103 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:93 *****
Symptom: 0.000000 0.000000 -1.555480 -1.684100 -2.016540 -1.883410 0.123350 0.189900
Faulty group=2 Most suitable rule no=4
0.023 0.000 0.000 0.000 0.110 0.047 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.003 0.000 0.000
***** data no:94 *****
Symptom: 0.000000 0.000000 -0.322110 -0.673680 -1.065050 -0.823240 0.084250 0.000720
Faulty group=2 Most suitable rule no=0
0.315 0.000 0.000 0.000 0.002 0.173 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:95 *****
Symptom: 0.000000 0.000000 -0.836640 -0.946230 -1.210370 -0.967900 0.082230 -0.008180
Faulty group=2 Most suitable rule no=0
0.193 0.000 0.000 0.001 0.011 0.131 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:96 *****
Symptom: 0.000000 0.000000 -2.402290 -2.751490 -3.436570 -2.941590 -0.147040 -0.025610
Faulty group=2 Most suitable rule no=4
0.000 0.006 0.000 0.000 0.679 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:97 *****
Symptom: 0.000000 0.000000 -0.750190 -0.785090 -0.817170 -0.317540 -0.162150 -0.030600
Faulty group=2 Most suitable rule no=0
0.337 0.000 0.001 0.003 0.002 0.126 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:98 *****
Symptom: 0.000000 0.000000 -0.921360 -1.084740 -1.454920 -0.991940 0.054680 -0.077580
Faulty group=2 Most suitable rule no=0
0.146 0.000 0.004 0.004 0.017 0.137 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:99 *****
Symptom: 0.000000 0.000000 -2.013540 -2.382890 -3.016290 -2.438000 0.231070 0.099130
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.387 0.011 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.007 0.000 0.000
***** data no:100 *****
Symptom: 0.000000 0.000000 -1.228330 -1.767290 -2.406190 -1.867960 -0.248650 -0.009920
Faulty group=2 Most suitable rule no=4
0.017 0.000 0.000 0.000 0.110 0.067 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:101 *****
Symptom: 0.000000 0.000000 -0.724450 -0.991970 -1.173630 -1.040530 -0.318990 0.011370
Faulty group=2 Most suitable rule no=0

0.180 0.000 0.000 0.000 0.010 0.115 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:102 *****
Symptom: 0.000000 0.000000 -1.133750 -1.597240 -1.931670 -1.609930 0.083350 -0.025850
Faulty group=2 Most suitable rule no=5
0.046 0.001 0.001 0.000 0.067 0.084 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.002 0.000 0.000
***** data no:103 *****
Symptom: 0.000000 0.000000 -0.210020 0.076260 0.138510 0.119920 0.020320 -0.030630
Faulty group=2 Most suitable rule no=0
0.816 0.000 0.000 0.008 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:104 *****
Symptom: 0.000000 0.000000 -0.010350 -0.888450 -0.977570 -0.956880 -0.232290 0.152350
Faulty group=2 Most suitable rule no=0
0.282 0.000 0.000 0.000 0.000 0.136 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:105 *****
Symptom: 0.000000 0.000000 -0.734540 -1.130000 -1.524650 -1.256270 0.056840 0.050470
Faulty group=2 Most suitable rule no=5
0.130 0.000 0.000 0.000 0.019 0.134 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000
***** data no:106 *****
Symptom: 0.000000 0.000000 -2.300520 -2.739260 -3.335620 -2.925610 0.068910 -0.020000
Faulty group=2 Most suitable rule no=4
0.000 0.004 0.000 0.000 0.663 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:107 *****
Symptom: 0.000000 0.000000 -0.036060 -0.621130 -1.076410 -0.690130 0.058370 -0.065990
Faulty group=2 Most suitable rule no=0
0.371 0.000 0.005 0.008 0.000 0.208 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:108 *****
Symptom: 0.000000 0.000000 -0.945000 -0.835130 -1.471110 -1.156890 0.041520 0.236820
Faulty group=2 Most suitable rule no=0
0.141 0.000 0.000 0.000 0.015 0.135 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:109 *****
Symptom: 0.000000 0.000000 -1.994060 -1.949990 -2.751770 -2.297840 0.141650 0.080230
Faulty group=2 Most suitable rule no=4
0.002 0.000 0.000 0.000 0.281 0.023 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.004 0.000 0.000
***** data no:110 *****
Symptom: 0.000000 0.000000 -1.225560 -1.607170 -2.047190 -1.769260 0.136510 -0.092390
Faulty group=2 Most suitable rule no=4
0.033 0.003 0.002 0.001 0.081 0.071 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.003 0.000 0.000
***** data no:111 *****
Symptom: 0.000000 0.000000 -6.573310 -8.816310 -11.672380 -9.440980 -0.412860 0.047220
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.849 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:112 *****
Symptom: 0.000000 0.000000 -11.359230 -14.119040 -17.644199 -14.934310 -0.105140 0.045950
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.950 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:113 *****
Symptom: 0.000000 0.000000 -16.356489 -18.125240 -21.497061 -19.153049 0.399550 0.161530
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.820 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:114 *****

Symptom: 0.000000 0.000000 -19.841999 -23.145170 -28.359119 -24.632839 -0.119550 0.220410
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.890 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:115 *****
Symptom: 0.000000 0.000000 -9.244330 -12.322830 -16.478670 -13.258270 -0.037380 0.130320
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.945 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:116 *****
Symptom: 0.000000 0.000000 -4.788520 -7.800340 -11.028380 -8.453730 -0.326240 0.184260
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.837 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:117 *****
Symptom: 0.000000 0.000000 -3.483090 -6.481120 -10.822640 -7.150500 0.058370 -0.018220
Faulty group=2 Most suitable rule no=4
0.000 0.006 0.000 0.000 0.975 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:118 *****
Symptom: 0.000000 0.000000 -7.314120 -10.381860 -15.802210 -11.566220 -0.052690 0.308820
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.881 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:119 *****
Symptom: 0.000000 0.000000 -15.954090 -19.860531 -26.714540 -21.608210 0.047350 0.296900
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.887 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:120 *****
Symptom: 0.000000 0.000000 -11.197980 -15.226360 -21.213461 -16.553961 0.042220 0.027500
Faulty group=2 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.977 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:121 *****
Symptom: 0.000000 0.000000 -3.257840 -4.109500 -5.735860 -4.219160 -0.211300 -0.147910
Faulty group=3 Most suitable rule no=4
0.000 0.046 0.000 0.000 0.884 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:122 *****
Symptom: 0.000000 0.000000 -4.181940 -5.150370 -6.587420 -5.495860 -0.432190 0.028200
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.848 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:123 *****
Symptom: 0.000000 0.000000 -5.292940 -5.875990 -7.032370 -6.395030 -0.536610 0.243430
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.755 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:124 *****
Symptom: 0.000000 0.000000 -2.769920 -4.161920 -6.028560 -4.668230 -0.292720 0.015650
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.829 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:125 *****
Symptom: 0.000000 0.000000 -3.855770 -4.930570 -6.540500 -5.308590 -0.388960 0.021740
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.864 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:126 *****
Symptom: 0.000000 0.000000 -6.494030 -7.454220 -9.118270 -7.982320 -0.805220 0.034500
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.723 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:127 *****
Symptom: 0.000000 0.000000 -3.585900 -5.072560 -7.268250 -5.074540 -0.538190 -0.010680
Faulty group=3 Most suitable rule no=4
0.000 0.003 0.000 0.000 0.818 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:128 *****
Symptom: 0.000000 0.000000 -4.352110 -5.885300 -8.305460 -6.276970 -0.322180 -0.037700
Faulty group=3 Most suitable rule no=4
0.000 0.011 0.000 0.000 0.881 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:129 *****
Symptom: 0.000000 0.000000 -6.898720 -8.304430 -10.646770 -8.881290 -0.335210 0.179390
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.835 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:130 *****
Symptom: 0.000000 0.000000 -5.470390 -7.221860 -9.714920 -7.822310 -0.718290 0.030060
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.753 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:131 *****
Symptom: 0.000000 0.000000 -6.029960 -7.910370 -11.012860 -8.424640 -0.682890 -0.121400
Faulty group=3 Most suitable rule no=4
0.000 0.031 0.000 0.000 0.741 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:132 *****
Symptom: 0.000000 0.000000 -8.025540 -9.971750 -12.784620 -10.703990 -0.997390 0.054790
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.655 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:133 *****
Symptom: 0.000000 0.000000 -11.269460 -12.615250 -15.073060 -13.584730 -1.196550 0.323730
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.536 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:134 *****
Symptom: 0.000000 0.000000 -5.565500 -8.182280 -11.687980 -9.021900 -0.763930 0.030590
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.738 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:135 *****
Symptom: 0.000000 0.000000 -7.465360 -9.721870 -12.943410 -10.487560 -0.860160 0.051640
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.701 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:136 *****
Symptom: 0.000000 0.000000 -12.521800 -14.367960 -17.513250 -15.385130 -1.369350 0.109620
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.524 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:137 *****
Symptom: 0.000000 0.000000 -6.694270 -9.761990 -14.386680 -10.310560 -0.914230 0.009240
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.693 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:138 *****
Symptom: 0.000000 0.000000 -8.243460 -11.337790 -16.108170 -12.292350 -0.793240 0.002180
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.735 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:139 *****
Symptom: 0.000000 0.000000 -13.133030 -15.905230 -20.399031 -17.079710 -0.901490 0.259650
Faulty group=3 Most suitable rule no=4

0.000 0.000 0.000 0.000 0.639 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:140 *****
Symptom: 0.000000 0.000000 -10.550370 -13.787540 -18.496201 -14.933610 -1.187930 0.110040
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.582 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:141 *****
Symptom: 0.000000 0.000000 -3.038820 -4.101650 -5.316820 -4.418290 -0.788300 0.035270
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.729 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:142 *****
Symptom: 0.000000 0.000000 -4.015950 -5.100120 -6.355390 -5.414390 -0.482100 -0.001910
Faulty group=3 Most suitable rule no=4
0.000 0.001 0.000 0.000 0.839 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:143 *****
Symptom: 0.000000 0.000000 -0.210020 0.076260 0.142330 0.103330 -0.264090 -0.030630
Faulty group=3 Most suitable rule no=0
0.752 0.000 0.000 0.008 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:144 *****
Symptom: 0.000000 0.000000 -2.502030 -4.467820 -6.195400 -4.909190 -0.608070 0.168300
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.628 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:145 *****
Symptom: 0.000000 0.000000 -3.992370 -5.415540 -7.241960 -5.898980 -0.414240 0.082420
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.838 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:146 *****
Symptom: 0.000000 0.000000 -6.590470 -7.729840 -9.437250 -8.273580 -0.590680 0.028090
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.796 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:147 *****
Symptom: 0.000000 0.000000 -2.554100 -4.875640 -8.181640 -5.457010 -0.224280 -0.042100
Faulty group=3 Most suitable rule no=4
0.000 0.011 0.000 0.000 0.777 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:148 *****
Symptom: 0.000000 0.000000 -4.442970 -6.079110 -9.364330 -6.953340 -0.335290 0.260820
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.811 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:149 *****
Symptom: 0.000000 0.000000 -7.238000 -8.666450 -11.741210 -9.598100 -0.329830 0.152450
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.845 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:150 *****
Symptom: 0.000000 0.000000 -5.669060 -7.666560 -10.565520 -8.401650 -0.334940 -0.044430
Faulty group=3 Most suitable rule no=4
0.000 0.013 0.000 0.000 0.875 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:151 *****
Symptom: 0.000000 0.000000 -5.798620 -7.813190 -10.284620 -8.436440 -1.257610 0.047220
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.572 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:152 *****

Symptom: 0.000000 0.000000 -7.814200 -9.770630 -12.196520 -10.411030 -0.953300 0.022020
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.677 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:153 *****
Symptom: 0.000000 0.000000 -4.261990 -4.474110 -5.261790 -4.768840 -0.738120 0.017410
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.750 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:154 *****
Symptom: 0.000000 0.000000 -5.286850 -8.540900 -12.084480 -9.363400 -0.983840 0.200220
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.627 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:155 *****
Symptom: 0.000000 0.000000 -7.887870 -10.558190 -14.105440 -11.454630 -0.885310 0.114350
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.678 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:156 *****
Symptom: 0.000000 0.000000 -12.681900 -14.790570 -18.115641 -15.848260 -1.156040 0.124250
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.589 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:157 *****
Symptom: 0.000000 0.000000 -5.218830 -9.451240 -15.742680 -10.537490 -0.601160 -0.018220
Faulty group=3 Most suitable rule no=4
0.000 0.005 0.000 0.000 0.795 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:158 *****
Symptom: 0.000000 0.000000 -8.304880 -11.860920 -18.045870 -13.311230 -0.712090 0.308820
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.684 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:159 *****
Symptom: 0.000000 0.000000 -13.496440 -16.705830 -22.495840 -18.302259 -0.895610 0.248760
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.643 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:160 *****
Symptom: 0.000000 0.000000 -10.740060 -14.591570 -20.311230 -15.955210 -0.806390 0.027500
Faulty group=3 Most suitable rule no=4
0.000 0.000 0.000 0.000 0.725 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:161 *****
Symptom: 0.000000 0.000000 4.844380 27.057619 -0.280230 7.917820 0.165980 -0.187670
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.803 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:162 *****
Symptom: 0.000000 0.000000 5.304220 21.805510 -0.109560 7.279900 0.038800 -0.051600
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.935 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:163 *****
Symptom: 0.000000 0.000000 0.541150 10.730340 -6.475050 0.583530 -0.536610 0.163130
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.513 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:164 *****
Symptom: 0.000000 0.000000 5.643580 29.538080 -0.159540 8.362840 0.084250 -0.044090
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.907 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:165 *****
Symptom: 0.000000 0.000000 5.702940 25.330250 0.145780 8.322100 0.082230 -0.053040
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.909 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.046
***** data no:166 *****
Symptom: 0.000000 0.000000 5.767470 17.463289 -0.127920 7.030280 -0.147040 -0.130780
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.871 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:167 *****
Symptom: 0.000000 0.000000 5.975530 33.380749 0.173750 10.848220 -0.162150 -0.090370
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.864 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.053
***** data no:168 *****
Symptom: 0.000000 0.000000 6.400730 29.555880 -0.089860 10.175680 0.054680 -0.137410
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.909 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:169 *****
Symptom: 0.000000 0.000000 6.755850 22.010330 -0.105920 9.036670 0.231070 -0.021260
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.884 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:170 *****
Symptom: 0.000000 0.000000 6.836800 25.000481 -0.352730 9.352540 -0.248650 -0.109900
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.780 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:171 *****
Symptom: 0.000000 0.000000 4.804840 24.788231 -0.188120 6.895300 -0.318990 -0.012540
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.834 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:172 *****
Symptom: 0.000000 0.000000 5.107300 19.862930 -0.262060 6.516200 0.083350 -0.073710
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.865 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:173 *****
Symptom: 0.000000 0.000000 -0.148620 10.106330 -6.976590 0.142040 -0.832930 -0.030630
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.647 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:174 *****
Symptom: 0.000000 0.000000 6.086940 30.647051 -0.091520 8.641600 -0.232290 0.120430
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.859 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:175 *****
Symptom: 0.000000 0.000000 6.140540 25.188280 -0.041380 8.429950 0.056840 -0.013400
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.963 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:176 *****
Symptom: 0.000000 0.000000 6.004580 18.369011 -0.123750 7.352030 0.068910 -0.132190
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.895 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:177 *****
Symptom: 0.000000 0.000000 6.613530 39.917099 -0.285450 11.687010 0.058370 -0.113770
Faulty group=4 Most suitable rule no=6

0.000 0.000 0.000 0.000 0.000 0.000 0.854 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:178 *****
Symptom: 0.000000 0.000000 6.677720 33.654072 -0.177470 11.331270 0.041520 0.140800
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.884 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:179 *****
Symptom: 0.000000 0.000000 7.164970 25.526400 -0.046070 10.355350 0.141650 -0.064220
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.918 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:180 *****
Symptom: 0.000000 0.000000 7.237430 28.632030 -0.120800 10.747070 0.136510 -0.212260
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.851 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:181 *****
Symptom: 0.000000 0.000000 10.039280 54.152401 -0.339760 15.671260 0.165980 -0.227440
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.774 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:182 *****
Symptom: 0.000000 0.000000 10.718610 43.677029 -0.199380 14.531000 0.038800 -0.091500
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.893 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:183 *****
Symptom: 0.000000 0.000000 0.581030 20.823111 -13.666320 0.600200 -1.102270 0.163130
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.386 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:184 *****
Symptom: 0.000000 0.000000 11.141110 59.040359 -0.226620 16.815830 0.084250 -0.073970
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.876 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:185 *****
Symptom: 0.000000 0.000000 11.429260 50.446739 0.051320 16.452290 0.082230 -0.097910
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.925 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.016
***** data no:186 *****
Symptom: 0.000000 0.000000 8.376300 31.431141 -4.580570 10.202050 -0.429120 -0.175850
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:187 *****
Symptom: 0.000000 0.000000 12.228630 66.809677 0.072640 21.221140 -0.162150 -0.150130
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.877 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.022
***** data no:188 *****
Symptom: 0.000000 0.000000 12.960450 59.070438 -0.216250 20.201309 0.054680 -0.217160
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.845 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:189 *****
Symptom: 0.000000 0.000000 13.534170 43.928890 -0.304490 17.913750 0.231070 -0.121570
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.796 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:190 *****

Symptom: 0.000000 0.000000 13.618850 50.152100 -0.508090 18.777769 -0.248650 -0.189870
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.714 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:191 *****
Symptom: 0.000000 0.000000 9.762810 49.866261 -0.248460 14.002420 -0.318990 -0.048380
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.806 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:192 *****
Symptom: 0.000000 0.000000 10.208880 39.913898 -0.337720 13.143320 0.083350 -0.121570
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.828 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:193 *****
Symptom: 0.000000 0.000000 -0.107690 20.136391 -14.118440 0.158630 -1.496570 -0.030630
Faulty group=4 Most suitable rule no=7
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.453 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:194 *****
Symptom: 0.000000 0.000000 11.729860 61.441978 -0.163130 17.502939 -0.232290 0.072560
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.851 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:195 *****
Symptom: 0.000000 0.000000 12.111310 50.296520 -0.142520 16.846930 0.056840 -0.061320
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.915 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:196 *****
Symptom: 0.000000 0.000000 11.943340 36.704750 -0.282530 14.688590 0.068910 -0.212330
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.822 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:197 *****
Symptom: 0.000000 0.000000 12.969760 80.053963 -0.406100 23.520571 0.058370 -0.185450
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.795 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:198 *****
Symptom: 0.000000 0.000000 13.693870 67.202049 -0.329060 22.787600 0.041520 0.068800
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.858 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:199 *****
Symptom: 0.000000 0.000000 14.623660 50.763969 -0.270980 20.644550 0.141650 -0.184600
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.813 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:200 *****
Symptom: 0.000000 0.000000 14.581070 57.370811 -0.307750 21.616819 0.136510 -0.308170
Faulty group=4 Most suitable rule no=6
0.000 0.000 0.000 0.000 0.000 0.000 0.769 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
***** data no:201 *****
Symptom: 0.000000 0.000000 -99.000000 -50.000000 -50.000000 -50.000000 -50.000000 -99.000000
Faulty group=7 Most suitable rule no=26
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000

Appendix F Outputs of the original SOM for fault classification

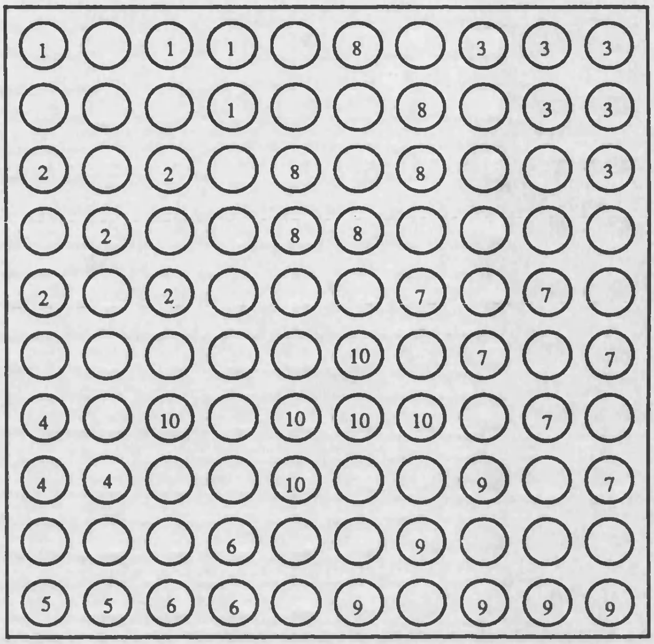


Figure a. Traning outputs of annulus pressure error patterns.

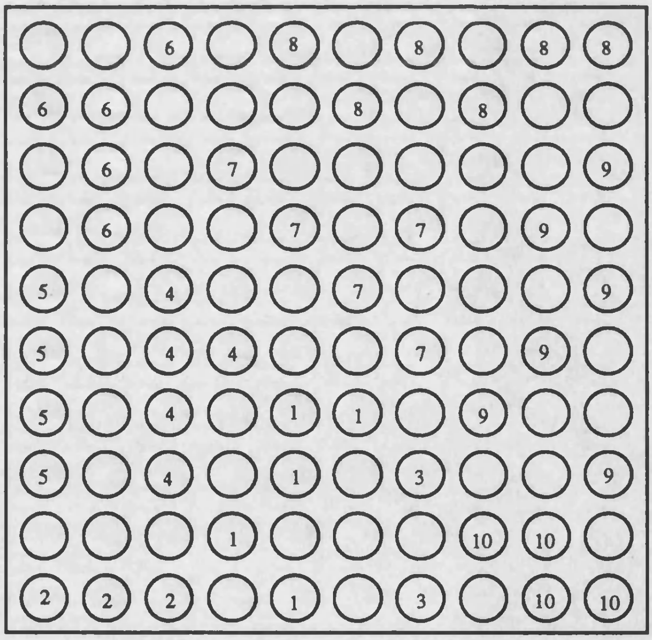


Figure b. Training outputs of piston pressure error patterns.

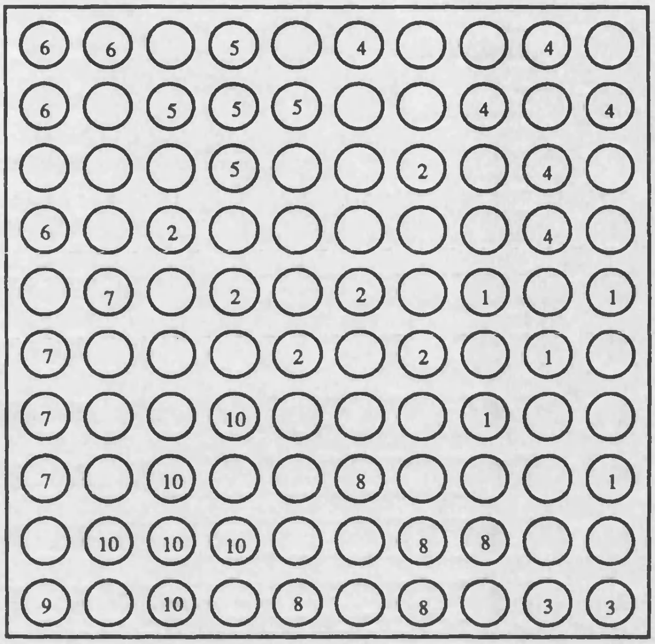


Figure c. Training outputs of system pressure error patterns.

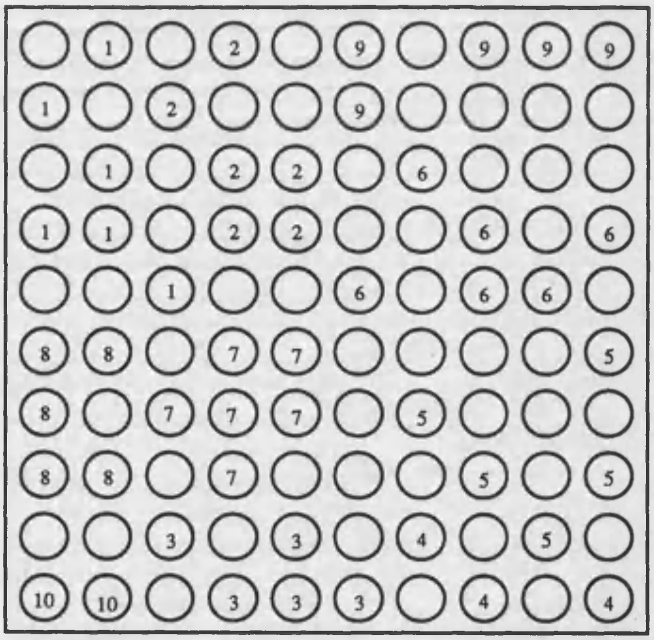
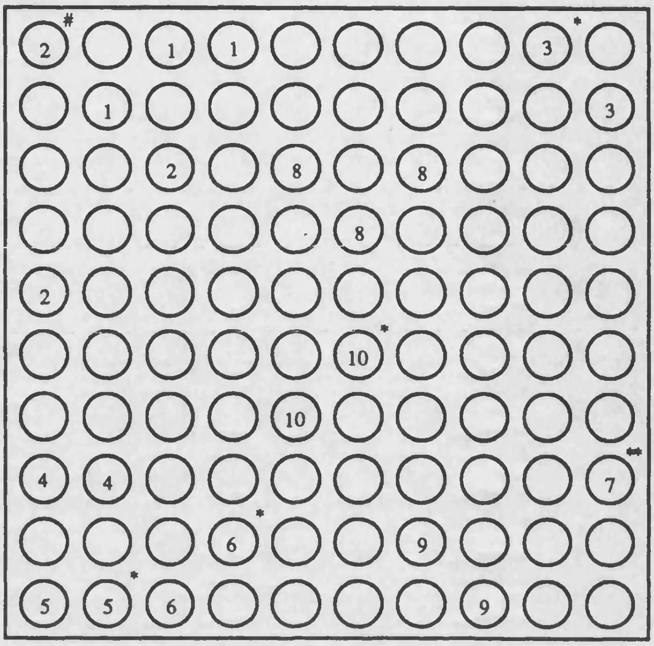
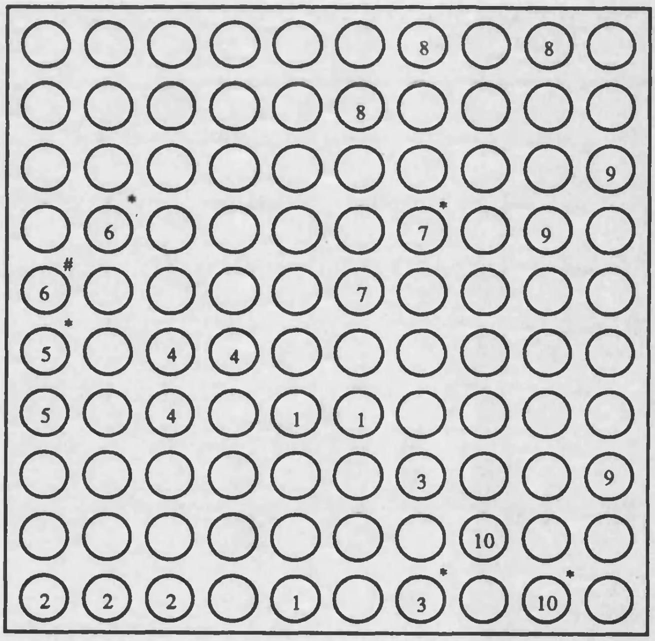


Figure d. Training outputs of piston displacement error patterns.



misclassified * output twice ** output three times

Figure e. Prediction outputs of annulus pressure error patterns.



misclassified * output twice

Figure f. Prediction outputs of piston pressure error patterns.

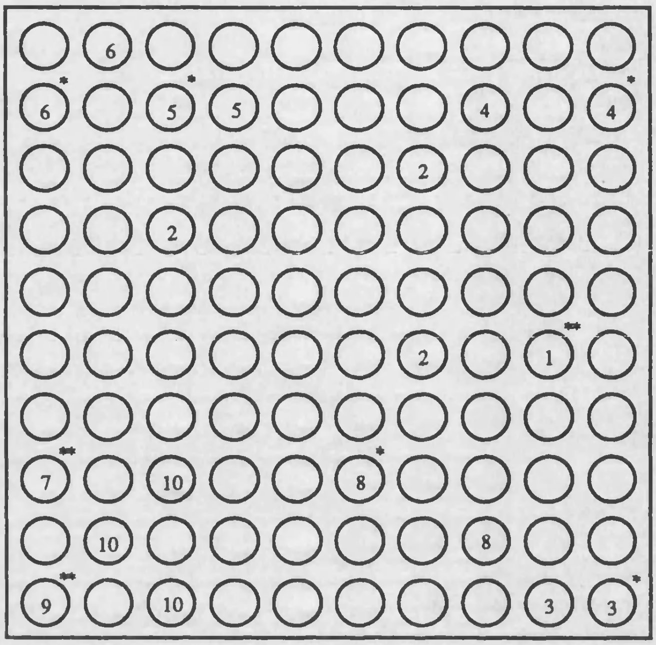


Figure g. Prediction outputs of system pressure error patterns.

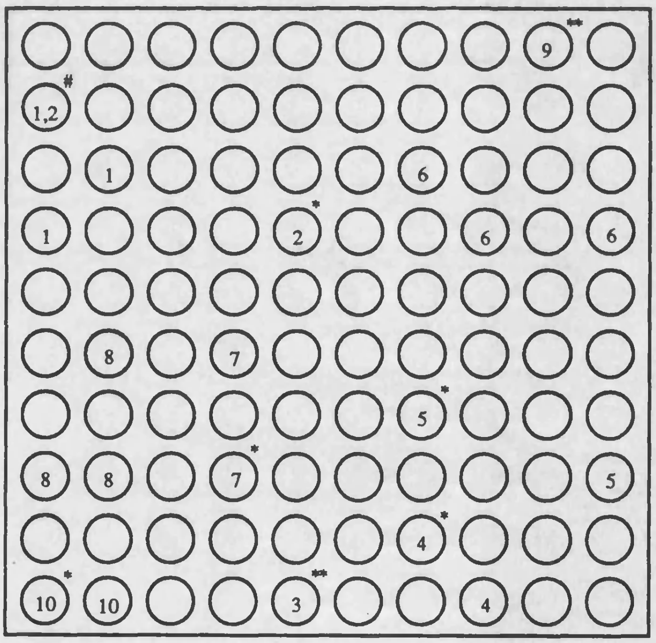


Figure h. Prediction outputs of piston displacement error patterns.

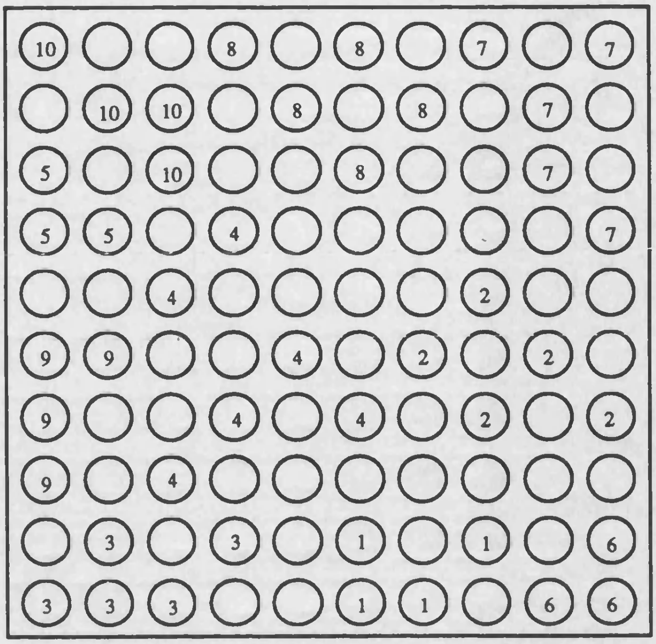
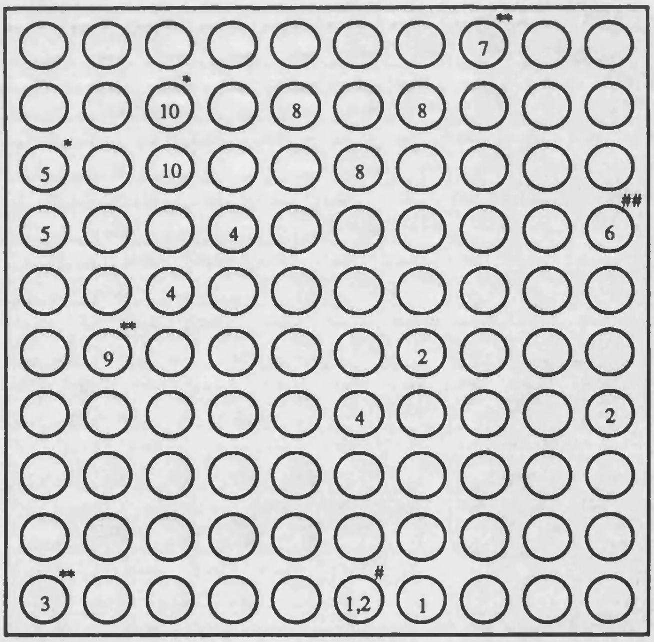


Figure i. Training outputs of combined error patterns.



no. 2 misclassified ## no. 6 misclassified three times

* output twice ** output three times

Figure j. Prediction outputs of combined error patterns.